AD-A034 542 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 15/3.1

ACQUISITION LOGIC FUNCTION ALGORITHM SPECIFICATION.(U)

JUN 76 E R WENTZ, H MOSTAFAVI, H M PEARCE F19628-76-C-0002

UNCLASSIFIED ESD-TR-76-244 NL

1 OF 2

AD
A034542

FG.

ESD-TR-76-244

ADA034542

2

ACQUISITION LOGIC FUNCTION ALGORITHM SPECIFICATION

FINAL REPORT

Report No. 5140-2

JUNE 1976

Prepared Under: CONTRACT NO. F19628-76-C-0002

Prepared For: MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lincoln Laboratory
P.O. Box 73
Lexington, Massachusetts 02173

Pepared By: SYSTEMS CONTROL, INC.
1801 Page Mill Road
Palo Alto, CA 94304

28

SYSTEMS CONTROL, INC.
1801 PAGE MILL ROAD
PALO ALTO, CALIFORNIA 94304

ESD-TR-76-244

TELEX: 348433

TELEPHONE (415)
494-1165

June 1976

ACQUISITION LOGIC FUNCTION ALGORITHM SPECIFICATION

SCI Report No. 5140-2

Purchase Order No. BB-307

Under Prime Contract No. F19628-76-C-0002

Contract Period: 21 April 1975 to 30 June 1976

Prepared for

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY
P. O. BOX 73
LEXINGTON, MASSACHUSETTS 02173

Approved for public release; distribution unlimited.

Approved by:

H. M. Pearce

H. M. Pearce
Program Manager

A J Tether

A. J. Tether
Vice President

Prepared by:
E. R. Wentz, Jr.
D. R. Morgan
H. Mostafavi
A. I. Cohen
H. M. Pearce

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Raymond L. Loiselle

Raymond L. Loiselle, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 18 ESD-TR-76-244
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): 6 Acquisition Logic Function Algorithm Specification,
5. TYPE OF REPORT & PERIOD COVERED: Final Report 4/21/75 - 6/30/76
6. PERFORMING ORG. REPORT NUMBER: SCI Report No. 5140-2
7. AUTHOR(s): 10 E. R./Wentz, Jr., H./Mostafavi, H. M./Pearce, D. R./Morgan, A. I./Cohen
8. CONTRACT OR GRANT NUMBER(s): 15 F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems Control, Inc. under Purchase Order No. BB-307 to M.I.T. Lincoln Laboratory
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 16 Project No. 8X363304D215
11. CONTROLLING OFFICE NAME AND ADDRESS: Ballistic Missile Defense Program Office, Department of the Army, 1300 Wilson Boulevard, Arlington, VA 22209
12. REPORT DATE: 11 June 1976
13. NUMBER OF PAGES: 119
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Electronic Systems Division, Hanscom AFB, Bedford, MA 01731 — 12 126p.
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

9 Final rept. 21 Apr 75 - 30 Jun 76,

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acquisition Logic Function
Bulk Filter
Terminal Defense
Ballistic Missile Defense
Known Object Recognition
Coincidence Detection
Simulation
Software

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report specifies the software design of an Acquisition Logic Function (ALF) for a monostatic radar, terminal defense, ballistic missile defense system. This report defines the data structure, data flow and program structure for the ALF simulation software. A separate report, "Acquisition Logic Function Analysis and Design," evaluates the algorithms specified in this report.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 207650

LB

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF TABLES

1.0 INTRODUCTION AND SUMMARY

1.1 INTRODUCTION

This report specifies the software design of an Acquisition Logic Function (ALF) for a monostatic radar, terminal defense, ballistic missile defense (BMD) system. A separate report evaluates the algorithms specified in this report.[1] This work was performed by Systems Control, Inc. (SCI) for the Massachusetts Institute of Technology Lincoln Laboratory (MIT-LL) as part of Purchase Order BB-307 under Prime Contract F19628-76-C-0002 during the period April 1975 through June 1976. The technical monitor for the work was Dr. G. Q. McDowell.

1.2 PURPOSE

MIT Lincoln Laboratory, in support of the Ballistic Missile Defense Advanced Technology Center (BMDATC), is engaged in evaluating bulk filter algorithms and acquisition logic for the terminal defense environment. The ultimate goal of the effort is to design, implement, and validate (in simulation) a complete Acquisition or Designation Function that would be capable of undergoing field tests with a phased array radar. The primary verification tool of the Acquisition Function will be a high-fidelity simulation of the radar environment as observed by a BMD system. This simulation (called the D&D Testbed) will exist on the BMD ARC computer facility.

The purpose of this report is to specify the software of the ALF simulation to be used on the ARC computer facility (CDC 7600) in conjunction with the D&D Testbed.

1.3 SPECIFICATION METHOD

This section describes the approach used in preparing the software specification. Although the ALF is intended for implementation on the CDC 7600, this report provides a computer-independent specification of the ALF software design.

The general form of this report is similar to the Milestone format used by the USAF Space and Missile Systems Organization. This format has proven itself to be very useful for the design specification of large software systems, since it clearly defines program structure, data structure, and data flow.

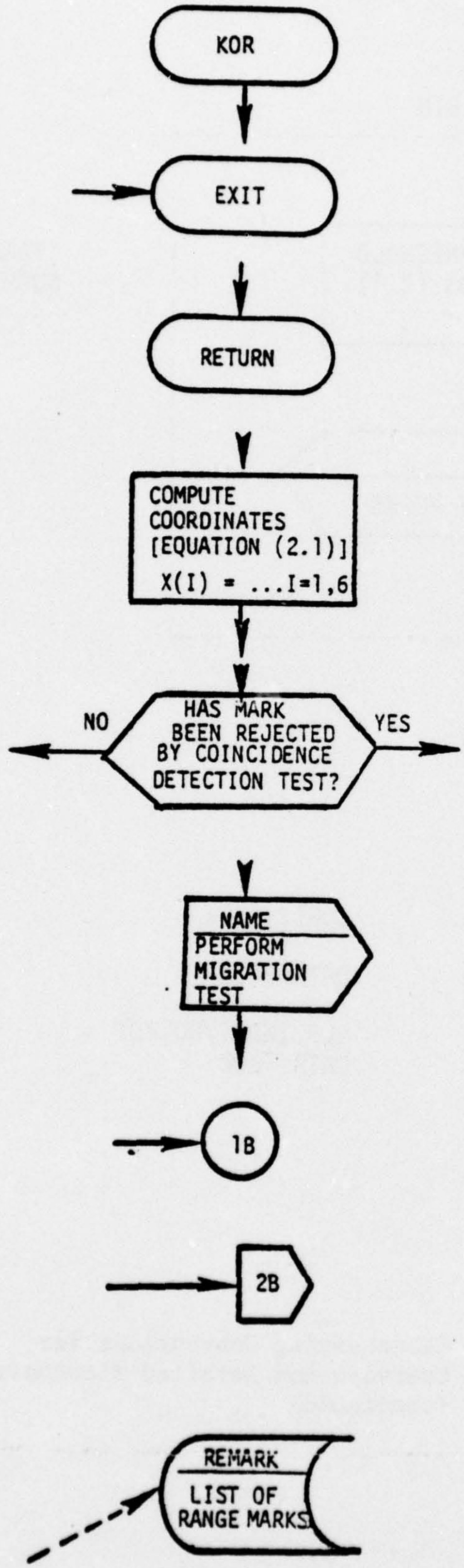
Flowchart Conventions

Figure 1.1 shows the flowcharting conventions used throughout this report for overview and detailed flowcharts. Overview flowcharts show the logic flow and data flow associated with the ALF and its subsystems. The detailed flowcharts show the logic flow for each of the subroutines defined in this report.

D&D Testbed/ALF Integration

The ALF software is designed to receive radar returns from the D&D Testbed. The origin of these returns is denoted throughout this report as the Matched Filter Processing Function. The Testbed and the ALF will be executed alternately, simulating the generation and processing of radar returns. Figure 1.2 shows an overview flowchart of this process. The ALF software design permits either open loop or closed loop operation with the Testbed, as desired. In the closed loop mode additional bursts are requested via the Waveform Request file as shown in Figure 1.2. Throughout the report this data is described as being sent to the Radar Scheduler Function, as would be the case in field implementation. In the open loop mode, returns are generated in a prearranged sequence, or recorded real returns are used.

The Testbed and the ALF programs and associated data require more core than is available in the small core memory of the CDC 7600. Therefore, some method must be provided to allow alternate execution and communication between the two programs. Two approaches have been suggested:[2]



START OF A PROGRAM

END OF ALF EXECUTION; RETURN TO BULK FILTER EXERCISER

END OF SUBROUTINE; RETURN TO CALLING PROGRAM

PROCESS, SUCH AS A COMPUTATION, ETC.

LOGIC TEST

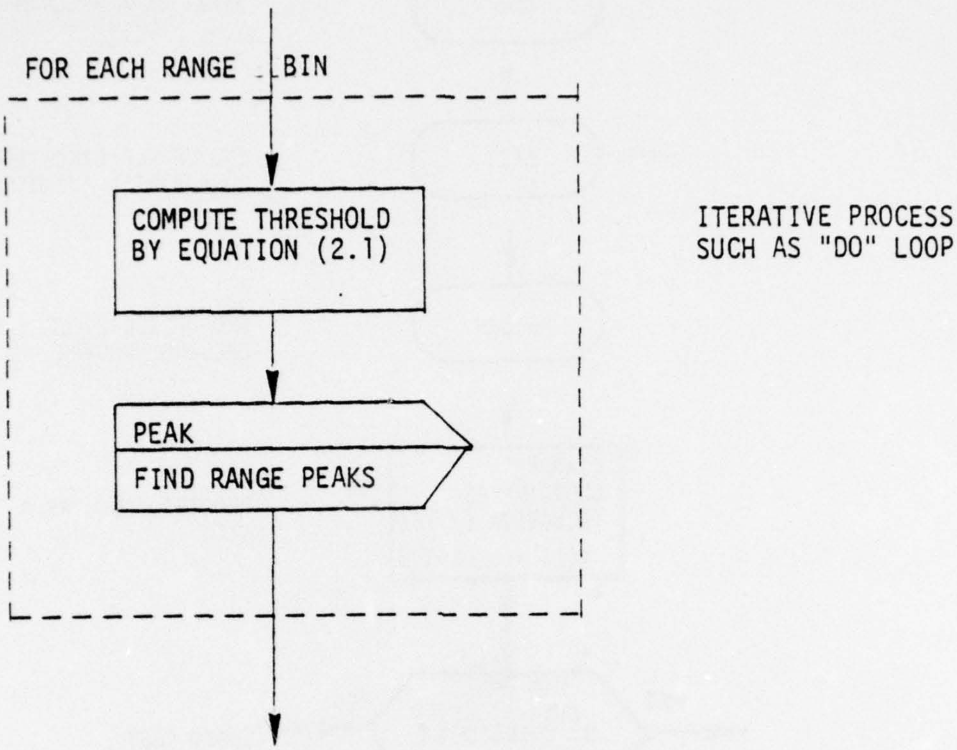
A SUBROUTINE

ON PAGE CONNECTOR (CONNECTS WITH ANOTHER 1B ON SAME PAGE).

OFF PAGE CONNECTOR (CONNECTS WITH 2B ON PAGE 2).

DATA FILE (OVERVIEW FLOWCHART ONLY; MAY BE SCM, LCM, OR DISC.)

Figure 1.1 Flowchart Conventions for Overview and Detailed Flowchart



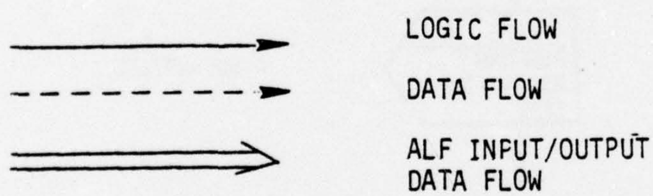


Figure 1.1 Flowcharting Conventions for Overview and Detailed Flowcharts (continued)

- Standard overlay processing (using LCM or disc)
- Control-point to control-point processing (each program appearing as a separate job to the operating system)

Development of an integration approach requires a tradeoff study involving run time versus data storage requirements. The computer-independent software specification presented in this report does not assume a specific integration approach.

An additional program called a Bulk Filter Exerciser[2] will be required for integration. This program will reformat data exchanged between the Testbed and ALF. If the overlay approach is used, an overlay executive program will also be required. Neither the Bulk Filter Exerciser nor the overlay executive are specified in this report. Their design is considered to be part of the development and integration phase.

Data Structure

Sections 2 through 9 of this report describe the ALF executive program and the subsystems which implement the component algorithms. Each section contains a subsection titled Data Specification which describes the arrays and variables that comprise the input and output data of the program or subsystem. The input/output data of each subsystem are listed in groups with a common source or destination to facilitate defining COMMON blocks for FORTRAN implementation. Data names use FORTRAN implicit typing: integer variables begin with letters I through N; other variables are real. For each item or array, the Data Specification subsections provide an engineering symbol, which is used in the text and equations, and a program symbol, which is used when discussing data flow, array sizes, etc.

A "Parameters" data file is associated with each algorithm, e.g., the MARKING ALGORITHM PARAMETERS file. In each case, such a file contains parameters used only in that algorithm.

Memory Management

The ALF data base (inputs, outputs, and working files) is described in this report as a set of arrays and variables. Because of the large size of the ALF data base, the use of mass memory will doubtless be required for some of the data. When implementing the ALF on the CDC 7600, data can reside in small core memory (SCM), large core memory (LCM), or on disc. When operating with the D&D Testbed, data can be saved from one ALF execution to the next by storing it in LCM or on disc. The location of each array in this memory heirarchy is not specified in this report and is considered to be one of the tasks in the development phase. Initial estimates for large array sizes are provided in Section 10.0, Memory Requirements.

Data Flow

Data flow is described at three levels:

- Between ALF and other functions
- Between ALF subsystems
- Between subroutines of a subsystem

The description of data flow is found in the Data Specification section for each subsystem and in the overview flowcharts and associated text.

Structured Programs

The ALF consists of a set of seven algorithms which are controlled by the executive program. Each algorithm is implemented by a set of subroutines with each subroutine performing a single functional task. The individual subroutines are described in sufficient detail to identify their basic structure and purpose. Top-down programming techniques are used so that logic flow proceeds in a unidirectional manner through each subroutine.

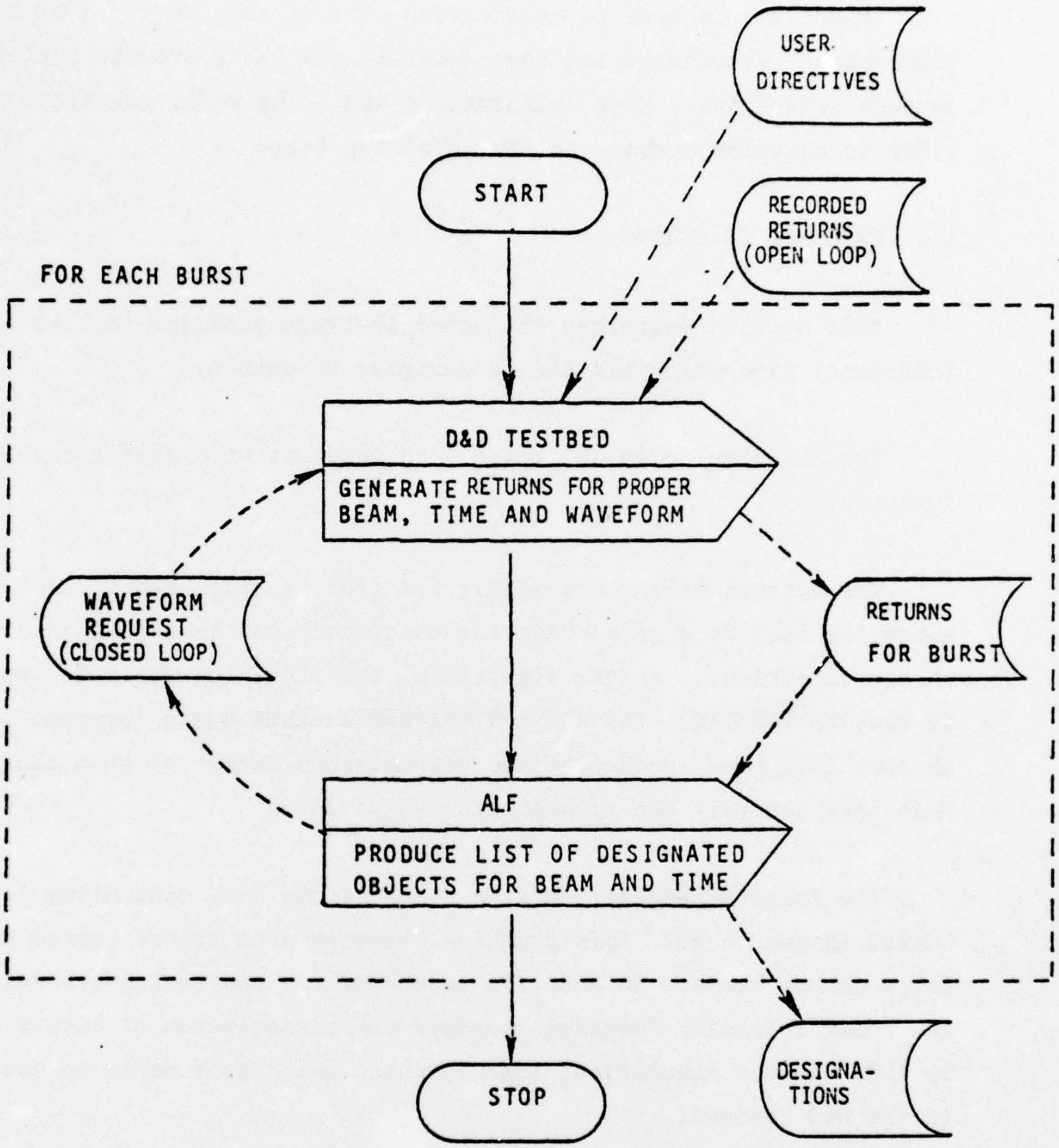


Figure 1.2 D&D Testbed/ALF Integration

Coding Details

No attempt is made to define each line of code required by each subroutine. Flowcharts and text describe the basic process performed in each subroutine. When computations are to be made, the flowcharts refer to equation numbers in the adjoining text.

1.4 EXTERNAL FUNCTIONS

This section describes the other software packages (called external functions) with which the ALF is designed to operate.

The ALF simulation is designed to interact with four external functions.

The Matched Filter Processing Function provides radar video and monopulse data in a given beam for specific burst transmission and threat conditions. In the simulation, this function would be implemented by the D&D Testbed. The ALF and the D&D Testbed would interact in either an open loop mode (predetermined transmission sequence) or a closed loop mode (ALF controls the transmission sequence).

The Radar Scheduler Function controls the time scheduling for the bursts in each beam. This function contains an a priori search pattern for a set of beams. In addition to scheduling the search transmissions, the Radar Scheduler Function controls the transmission of bursts requested by ALF. In the simulation, this function would presumably be contained in the D&D Testbed.

The Resource Allocation Function allocates BMD system resources (e.g., burst transmissions) among the BMD subsystems (e.g., ALF and Track Function). Future versions of ALF will include a Parameter Selection Algorithm for selecting among a set of designation sequences (see Section 4.0). This algorithm will interface with the Resource Allocation Function.

The Track Function controls radar tracking of the designated objects and periodically provides updated state estimates for those objects. This data is used by the Known Object Recognition Algorithm for removing redundant designations. If this function is not implemented in the simulation, the Known Object Recognition Algorithm will still remove redundant objects by comparing designation files for each beam.

1.5 OVERVIEW DISCUSSION

This section provides a summary description of the ALF component algorithms.

The framework used in designing the ALF is shown in Figure 1.3. The ALF is divided into seven component algorithms* (shown in heavy outlines). This division is somewhat arbitrary; however, it aids in defining the requirements of the ALF and the interfaces with the external functions: radar scheduler, resource allocator, and track function. The arrows in Figure 1.3 represent the flow of information between algorithms.

The ALF executive program determines the sequence of processing steps in the ALF and passes data, as required, to the various algorithms.

The Waveform Request Algorithm provides (1) the waveform parameters for each burst transmission and (2) the range, range-rate acceptance window for processing each beam. For the ALF design specified in this report, all of these parameters are calculated a priori and, hence, the Waveform Request Algorithm simply accesses a file containing the information.

The ALF Marking Algorithm processes radar returns supplied by the Matched Filter Processing Function. This algorithm locates peaks in range and range-rate that exceed the detection threshold.

*The ALF specified in this document includes the bulk filter-track initiation algorithm (i.e., the marking algorithm, the coincidence detection algorithm, and the designation sequence combining algorithm).

The <u>Coincidence Detection Algorithm</u> processes the list of marks from two burst waveforms to remove range ambiguities introduced by the waveform. The SCI coincidence detection algorithm is specified in this report.

The <u>Coherent Double Gating Algorithm</u> performs additional ambiguity removal and clutter rejection; it also provides initial state estimates in range and range-rate for the Track Function. The CDG Algorithm does not process angle measurements.

The <u>Parameter Selection Logic</u> executes the real-time selection of ALF parameters based on the measured environment. The current algorithm specifies a predetermined N-M:N designation sequence.

The <u>Object Beam Position Logic</u> computes the angular position of objects within the beam during the designation sequence.

The <u>Known Object Recognition (KOR) Algorithm</u> tests new designations against targets in track to prevent the initiation of redundant tracks on the same target. The KOR algorithm accomplishes both the "adjacent beam correlation" and the "designation-to-track correlation" operations.

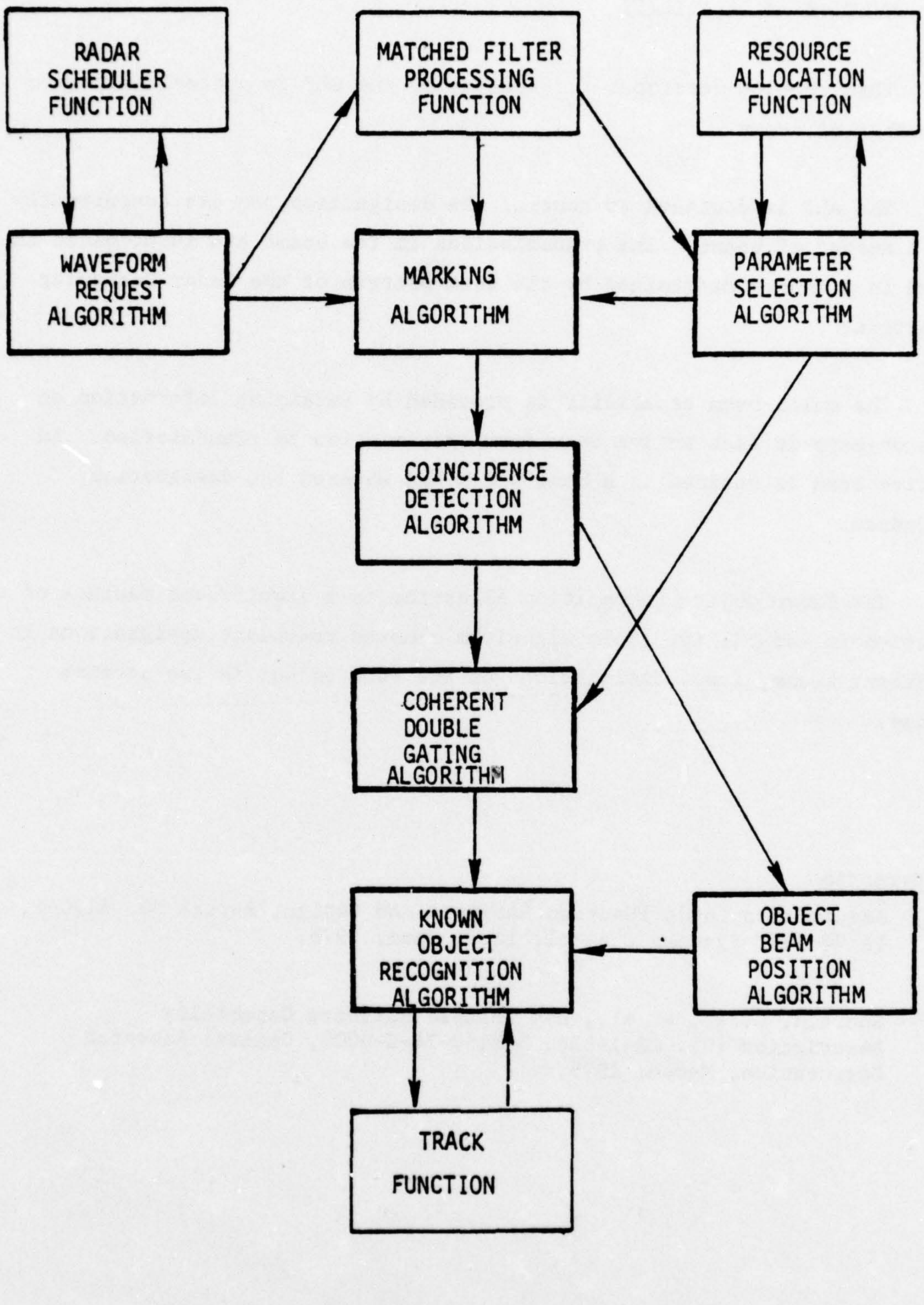


Figure 1.3 System Block Diagram Showing ALF Data Interfaces

1.6 MULTI-BEAM CAPABILITY

This section describes the ability of the ALF to process data from a number of beams.

The ALF is designed to control the designation process concurrently in a number of beams. The transmissions in the beams are interleaved in time in a manner determined by the scan pattern of the Radar Scheduler Function.

The multi-beam capability is provided by retaining information on the objects in each active beam from transmission to transmission. An active beam is defined as a beam which has entered the designation sequence.

The Known Object Recognition Algorithm is a significant feature of the multi-beam capability. This algorithm removes redundant designations in adjacent beams, i.e., designations on the same object in two or more beams.

References

[1] Acquisition Logic Function Analysis and Design, Report No. 5140-1, PA 76-319, Systems Control, Inc., June, 1976.

[2] Shortle, G. E., et al., D&D Testbed Software Capability Description (U), CR-1-626, DASG60-74-C-0009, General Research Corporation, March, 1975.

2.0 ALF EXECUTIVE PROGRAM

2.1 PURPOSE

The ALF executive program controls the execution of the ALF software. The executive calls each of the subroutines which implement ALF subsystems.

2.2 GENERAL DESCRIPTION

This section describes the sequence in which the ALF algorithms are executed and the data flow between algorithms.

The structure of the ALF and the basic purpose of each of its subsystems is described in Section 1.

In each pass through the ALF software, radar data is processed for one burst from one beam. Various data files are retained concerning the status of that beam; this data is used and updated when the next burst from that beam is processed.

The ALF is called by an executive program whenever radar data is available for a burst from a beam. The Matched Filter Processing Function provides the radar video and monopulse data for the beam. The beam number and other data which identify the source and nature of the radar returns are also provided. If this data exists on disc, the ALF executive reads this data and stores it in core memory (probably LCM).

The ALF design assumes that the Radar Scheduler Function will initially cause the Matched Filter Processing Function to provide radar and monopulse data on a predetermined schedule for each beam. This process simulates the transmission of search bursts. The purpose of a search burst is to determine whether there are any objects in a beam and, if so, to provide an initial set of marks for that beam.

Figure 2.1 is an overview flowchart for the ALF software.

If the first burst of a pair is being processed, Figure 2.1 shows the Marking Algorithm (MRKALG) is called to process the BURST 1 VIDEO file and to pick peaks in range and range-rate. The location of these marks is stored in the BURST 1 MARKS file. If there are any marks and if the burst was a search burst, then a second burst is requested by the Waveform Request Algorithm 2-3 msec after the search burst. If there are no marks on a search burst, no additional burst is requested, and the next transmission in that beam will be determined by the a priori schedule of the Radar Scheduler Function.

If the ALF is processing the second burst of a pair, processing follows the logic path shown on the right hand side of the first page of Figure 2.1. Marking is performed on the BURST 2 VIDEO file and the marks are stored in the BURST 2 MARKS file. The Coincidence Detection Algorithm then examines marks and video from both bursts to produce a list of COINCIDENCE DETECTION MARKS.

If the burst pair number equals or exceeds a parameter M (defined below) the Object Beam Position Algorithm (ANGALG) is called to compute angular positions of all objects whose marks passed coincidence detection. This data is stored with each mark in the COINCIDENCE DETECTION MARKS file.

If the burst pair being processed is the first pair of a designation sequence, a test is made to determine if any marks passed coincidence detection. If no marks passed, processing is terminated and the beam reverts to the search mode. If there are coincidence detection marks, the Parameter Selection Algorithm is called to determine an N-M:N designation sequence for the beam.* The Coherent Double Gating Algorithm (CDGALG) is called and initialized (for the set of coincidence detection marks) to form the CDG TRACK FILE. The Waveform Request Algorithm is then called to request the next burst pair 50 msec later. Once the designation sequence of N burst pairs has begun, burst pairs are requested rather than single bursts.

---

*In the CDG algorithm a strategy which requires N-M detections in a total of N burst pair transmissions is denoted by N-M:N.

Video data from successive transmissions are processed as described above up to the point marked 2A on Figure 2.1. At this point, the Coherent Double Gating Algorithm is called to update the object tracks from the previous transmission (Page 2 of Fig. 2.1). Those objects whose tracks pass the gating test are retained (in updated form) in the CDG TRACK FILE. Then if there are tracks remaining in the beam and the designation sequence is not completed, another burst pair is requested by the Waveform Request Algorithm. If there are no tracks remaining, the beam reverts to the search mode. If there are tracks remaining at the end of the designation sequence (after burst N), the Known Object Recognition Algorithm (KORALG) is called to remove redundant designations within the beam. The designations corresponding to the adjacent beams in DESIGNATIONS file are examined to remove any designations from the current beam which are redundant between beams. The KOR Algorithm also removes designations from the current beam which are indistinguishable from objects already in track. Those objects within the beam which pass the KOR test are passed to the Track Function in the DESIGNATIONS File.

Figure 2.2 shows the sequence of events for processing a single beam. Each horizontal sequence describes the generation and processing of returns for one burst in that beam. The sequences in Figure 2.2 are shown in the time order of occurrence (from the top down). These sequences depict each major logic path through the overview flowchart in Figure 2.1.

In Figure 2.2 a reference to burst (k,ℓ) means the ℓ^{th} burst (first or second) of burst pair k, where

$$1 \leq k \leq N.$$

Burst (1,1) is the search burst and is requested by the Radar Scheduler Function. The presence of marks on a search burst initiates the acquisition sequence, i.e., burst (1,2) is requested, as shown in the second sequence in Figure 2.2.

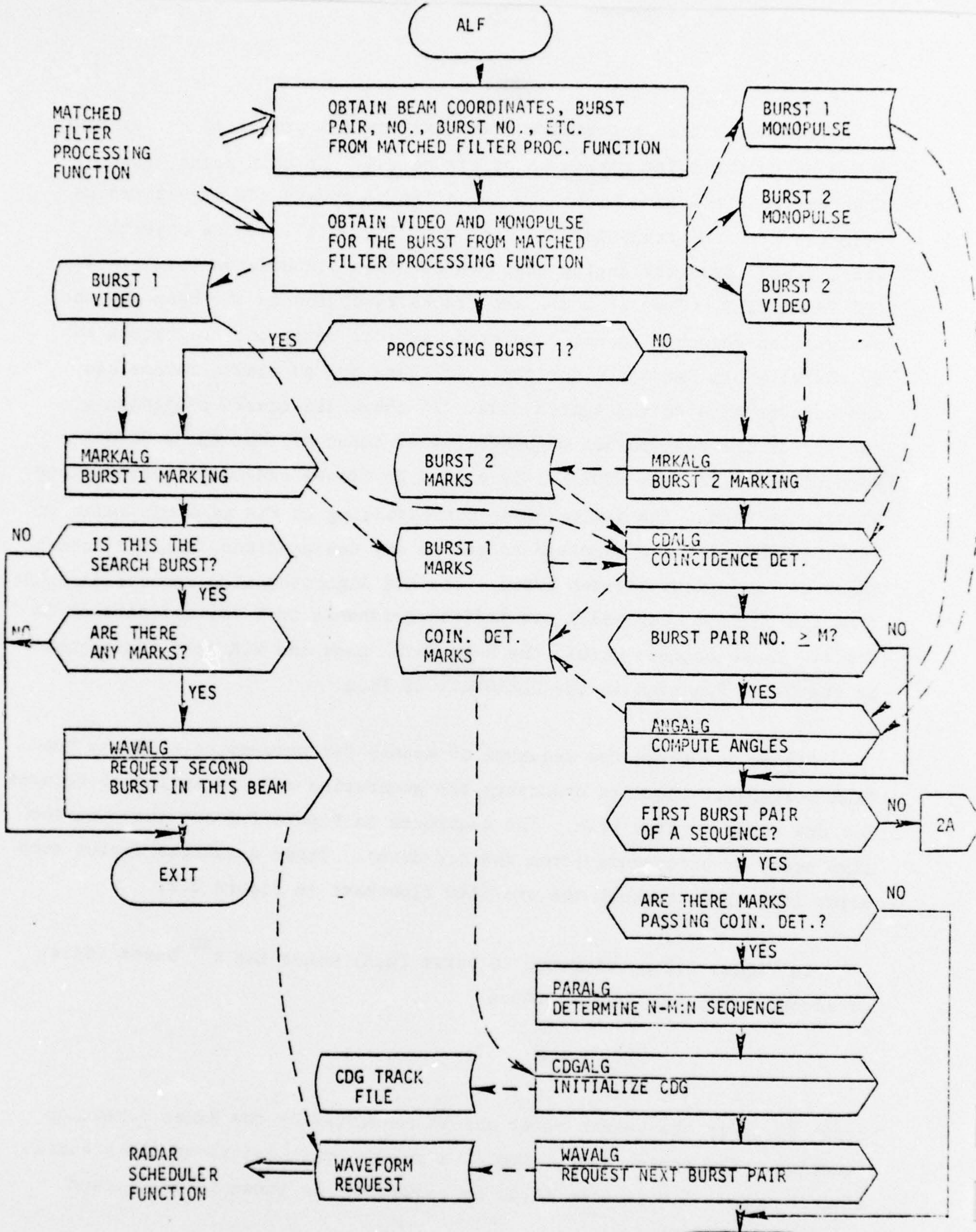


Figure 2.1 ALF Software Overview Flowchart

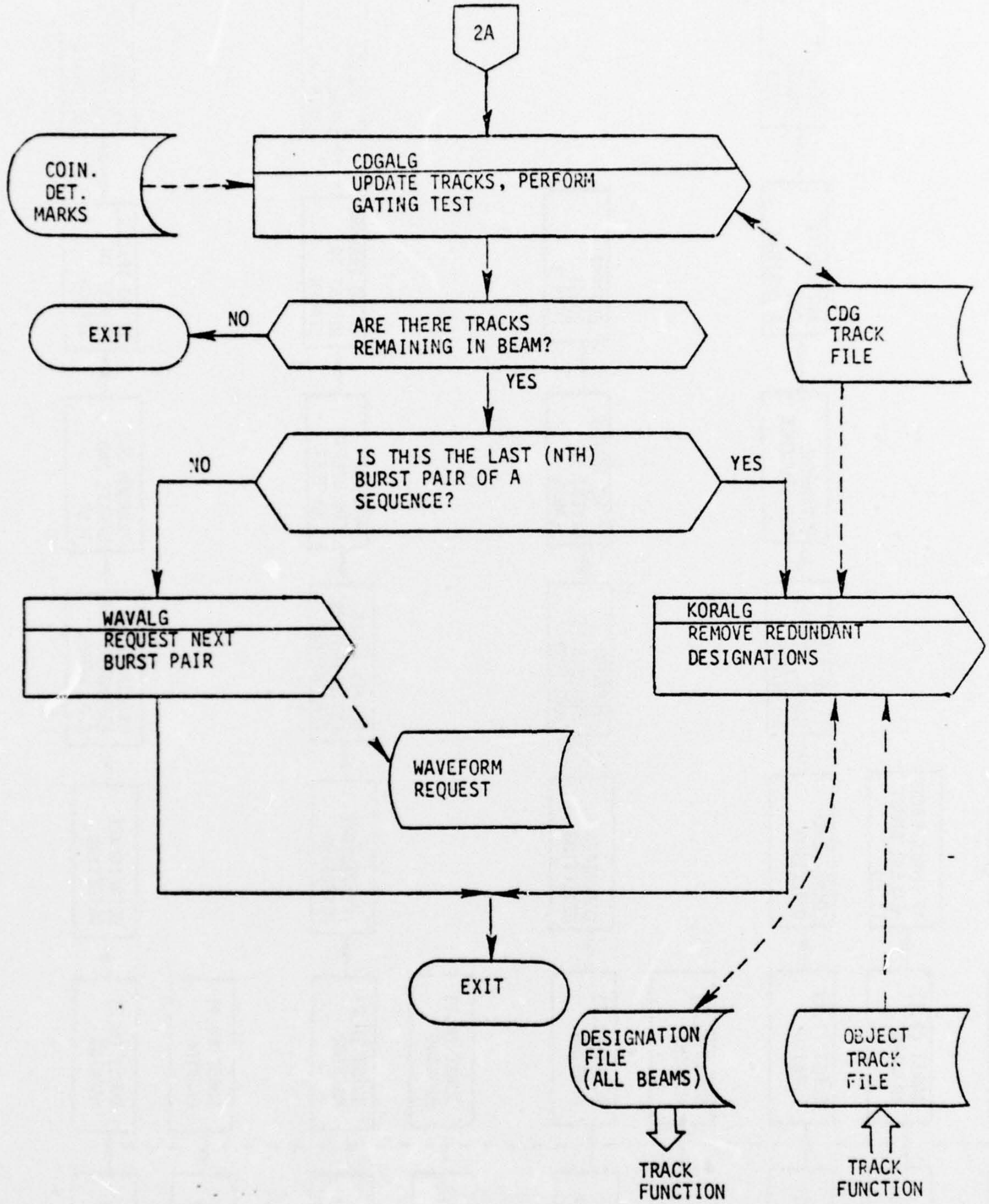


Figure 2.1 Software Overview Flowchart (Continued)

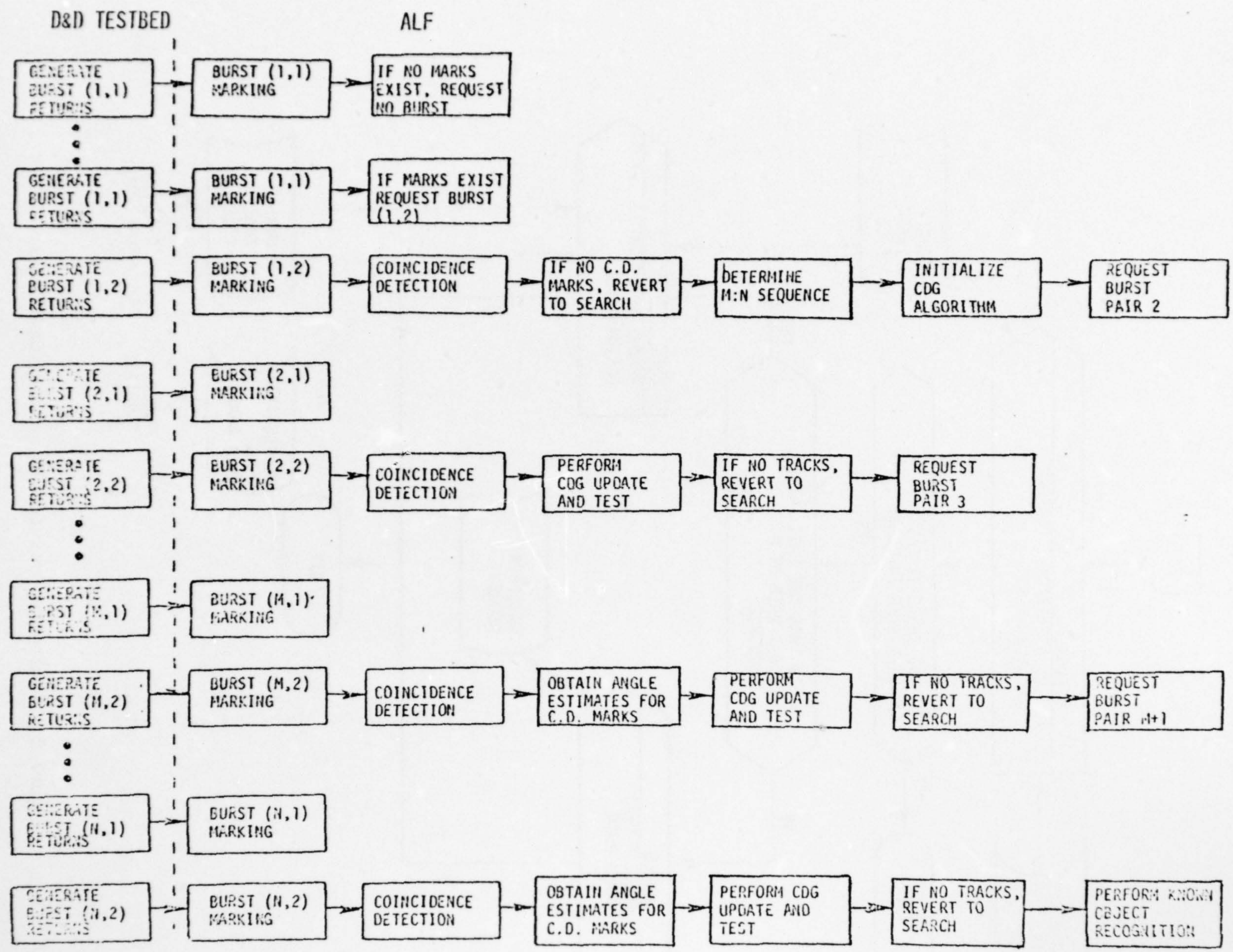


Figure 2.2 Sequence of Events for Acquisition Processing in a Single Beam

Burst transmissions for various beams are interleaved in time. Therefore the single beam processing shown in Figure 2.2 is interleaved with the processing of other beams, since bursts are processed in the order in which they are received.

The BEAM STATUS file contains information which describes the current status of each beam which is either undergoing or has completed designation processing. The file entry for a beam is updated by the ALF executve program each time a burst is processed for that beam.

2.3 DATA SPECIFICATION

This section discusses the data interchanged between the ALF executive and the external functions. A bookkeeping File (BEAM STATUS) is also discussed.

2.3.1 Input Data

The following data is input from the Matched Filter Processing Function:

Program Symbol	Engineering Symbol	Description
IU	-	Index for beam U coordinate
IV	-	Index for beam V coordinate
NBPAIR	-	Burst pair number for current burst
NBURST	-	Location of the current burst within a burst pair (NBURST = 1,2)
RS	R_S	Range corresponding to start of video data processing (m)
RE	R_E	Range corresponding to end of video data processing (m)
DELR	Δ	Range sample spacing (m)
DS	D_S	Range-rate corresponding to lowest Doppler channel for which video data is processed (m/s)
DE	D_E	Range-rate corresponding to highest Doppler channel for which video data is processed (m/s)
DELD	ζ	Range-rate sample spacing (Doppler channel spacing) (m/s)
VIDEO(K,I)	$r(k;i)$	Amplitude of video at range $R_S + (k-1)\Delta$ and range-rate $D_S + (i-1)\zeta$. Note that $k=1$ at R_S and $i=1$ at D_S.
UBST1(K,I)	$\Delta u(k,i)$	Monopulse measurement of the U coordinate; indexed like $r(k,i)$
VBST1(K,I)	$\Delta v(k,i)$	Monopulse measurement of the V coordinate; indexed like $r(k,i)$

The following data is input from the TRACK FUNCTION via the OBJECT TRACK file:

Program Symbol	Engineering Symbol	Description
NTRK	-	Number of objects in the array TRACKS
TRACKS(NTRK,I)	-	List of the most recent data on the objects in track i = 1 Range estimate (m) i = 2 Range-rate estimate (m/sec) i = 3 Variance of range estimate (m^2) i = 4 Covariance of range and range-rate estimates (m^2/sec) i = 5 Variance of the range-rate estimate (m^2/sec^2) i = 6 Angular coordinate U i = 7 Angular coordinate V i = 8 Time of estimation (msec)

The following data is input via the BEAM STATUS File:

ICUR	-	Index to array entries for current beam
IDES(ICUR)	-	Flag indicating designation status of beam IDES(ICUR) = 1 Designation complete IDES(ICUR) = 0 Designation incomplete
TBURST(ICUR)	-	Time of last burst transmission in the beam (msec)
IU(ICUR)	-	Index for beam U coordinate
IV(ICUR)	-	Index for beam V coordinate
NBPAIR(ICUR)	-	Number of burst pair being processed in the beam
NBURST(ICUR)	-	Number of burst being processed in the beam
NBEAM	-	Number of beams currently listed in the BEAM STATUS file

2.3.2 Output Data

The following data is output to the Radar Scheduler Function:

Program Symbol	Engineering Symbol	Description
NBREQ		Number of bursts requested (1 or 2)
TNEXT	-	Transmission time for next requested burst in the beam (typically 50 msec after current burst pair)
TSEP	-	Time separation between bursts in the requested burst pair
TIPS1	-	Interpulse spacing in the next requested burst (μsec)
TIPS2	-	Interpulse spacing in the second burst of requested burst pair (μsec)
IU	-	Index for beam U coordinate
IV	-	Index for beam V coordinate

The following data is output to the Track Function via the DESIGNATIONS file:

NDES	-	Number of objects in the array DESIG
DESIG(NDES,I)	-	List of objects designated in the beams already processed in the current scan of the group of beams. i = 1 Range estimate (m) i = 2 Range-rate estimate (m^2) i = 3 Variance of range estimate (m^2) i = 4 Covariance of range and range-rate estimates (m^2/sec) i = 5 Variance of range-rate estimate (m^2/sec^2) i = 6 Angular coordinate U i = 7 Angular coordinate V i = 8 U index for beam of designation i = 9 V index for beam of designation

The updated BEAM STATUS file is output by the ALF executive.

2.4 MATHEMATICAL RELATIONSHIPS

The mathematical relationships embodied in the ALF algorithms are described in the sections which specify the individual algorithms. The executive program provides the logic for sequential execution of the algorithms.

3.0 WAVEFORM REQUEST ALGORITHM

3.1 PURPOSE

The Waveform Request Algorithm requests the Radar Scheduler Function to transmit specific waveforms required for bulk filtering. It also provides parameters required by other algorithms for processing the radar return signal.

The WAVEFORM REQUEST file is generated only when the D&D Testbed/ALF is operating in the closed loop mode, i.e., only when the ALF determines what burst will be transmitted next.

3.2 GENERAL DESCRIPTION

Figure 3.1 is the overview flowchart for the Waveform Request Algorithm (WAVALG). The BEAM STATUS file indicates which beam has just been processed, the transmission time of the received burst, and the burst pair and burst number. From this data, the Waveform Request Algorithm computes the transmission time and interpulse spacing for the next burst or burst pair in the beam. These computed burst parameters are stored in the WAVEFORM REQUEST file (along with the beam indices, IUREQ and IVREQ) for use by the Radar Scheduler Function.

The Waveform Request Algorithm chooses a different interpulse spacing from a prestored table for each successive requested burst. The pulse spacing is varied to provide additional range ambiguity removal in the Coherent Double Gating Algorithm.

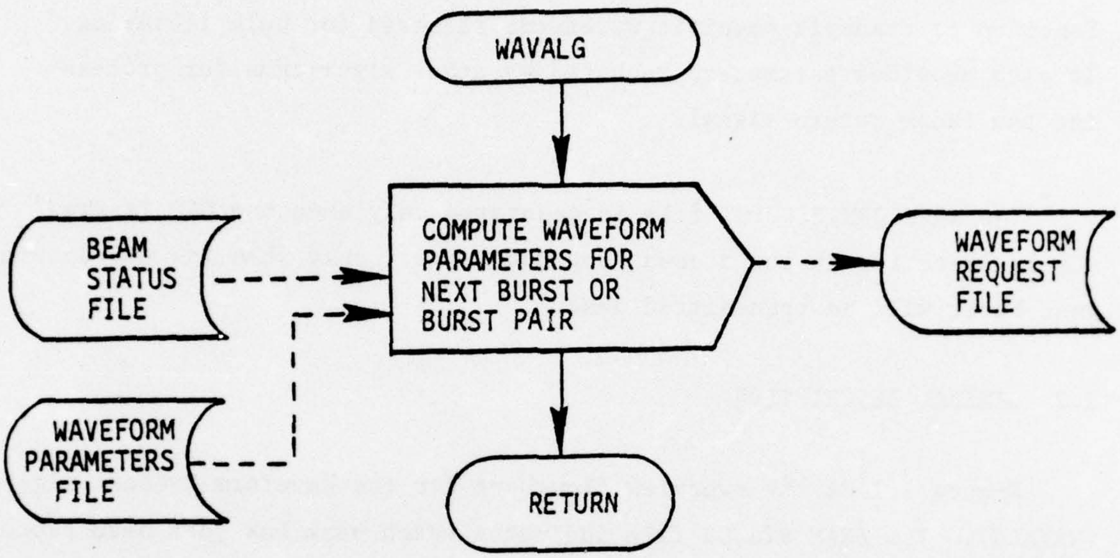


Figure 3.1 Waveform Request Algorithm Overview Flowchart

3.3 DATA SPECIFICATION

3.3.1 Input Data

The following data is input from the ALF Executive via the BEAM STATUS File:

Program Symbol	Engineering Symbol	Description
ICUR	-	Index to array entries for current beam
IDES(ICUR)	-	Flag indicating designation status of beam IDES(ICUR) = 1 Designation complete IDES(ICUR) = 0 Designation incomplete
TBURST(ICUR)	-	Time of last burst transmission in the beam (msec)
IU(ICUR)	-	Index for beam U coordinate
IV(ICUR)	-	Index for beam V coordinte
NBPAIR(ICUR)	-	Number of burst pair being processed in the beam
NBURST(ICUR)	-	Number of burst being processed in the beam
NBEAM	-	Number of beams currently listed in the BEAM STATUS file.

The following data is input via the WAVEFORM PARAMETERS file:

Program Symbol	Engineering Symbol	Description
VALIPS(I,J)	-	Prestored values of interpulse spacing (μsec) I Burst pair number J Burst number
TB	T_B	Time between bursts (msec)
TP	T_P	Time between burst pairs (msec)
BEAMS(IU,IV,L)	-	List of range, range-rate processing limits for each beam in the search sector IU Beam U index IV Beam V index

R_{SW}	L = 1	Range corresponding to closest edge of acquisition band in the beam (m)
R_{EW}	L = 2	Range corresponding to the furthest edge of acquisition band in the beam (m)
D_S	L = 3	Range-rate corresponding to the lowest Doppler channel to be processed for the beam (m/sec)
D_E	L = 4	Range-rate corresponding to the highest Doppler channel to be processed for the beam (m/sec)

3.3.2 Output Data

The following data is output to the Radar Scheduler Function via the WAVEFORM REQUEST File:

Program Symbol	Engineering Symbol	Description
NBREQ	-	Number of bursts requested (1 or 2)
TNEXT	-	Transmission time for next requested burst in the beam (msec)
TSEP	T_B	Time separation between bursts in the requested burst pair (msec)
TIPS1	-	Interpulse spacing in the next requested burst (μsec)
TIPS2	-	Interpulse spacing in the second burst of requested burst pair (μsec)
IUREQ	-	Index for beam U coordinate
IVREQ	-	Index for beam V coordinate

3.3.3 Parameters

Table 3.1 shows recommended values of interpulse spacing for three successive burst pairs in a beam, where the nominal pulse spacing is 5.0 μsec. Data is stored in array VALIPS(I,J). By varying the pulse

Table 3.1

RECOMMENDED INTERPULSE SPACING VALUES (nominal=5.0 μsec)

BURST PAIR SEQUENCE NUMBER	BURST NUMBER WITHIN A BURST PAIR 1	2
1	5.0 sec	4.7 sec
2	5.2	4.9
3	5.1	4.8

spacing on successive burst pairs, the Coherent Double Gating Algorithm can reject ambiguities which have passed through the Coincidence Detection Algorithm. For designation sequences longer than 3 burst pairs, the sequence of interpulse spacings in Table 3.1 is repeated as many times as necessary.

Other parameters are as follows:

T_B Time between bursts (nominally 2 to 3 msec)

T_P Time between burst pairs (nominally 50 msec)

The two parameters T_B and T_P are defined as shown in Figure 3.2.

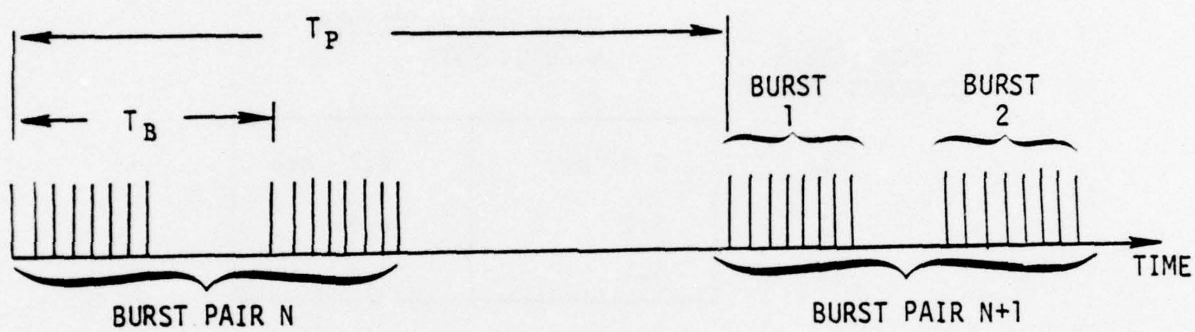


Figure 3.2 Burst Transmission Timing Definitions

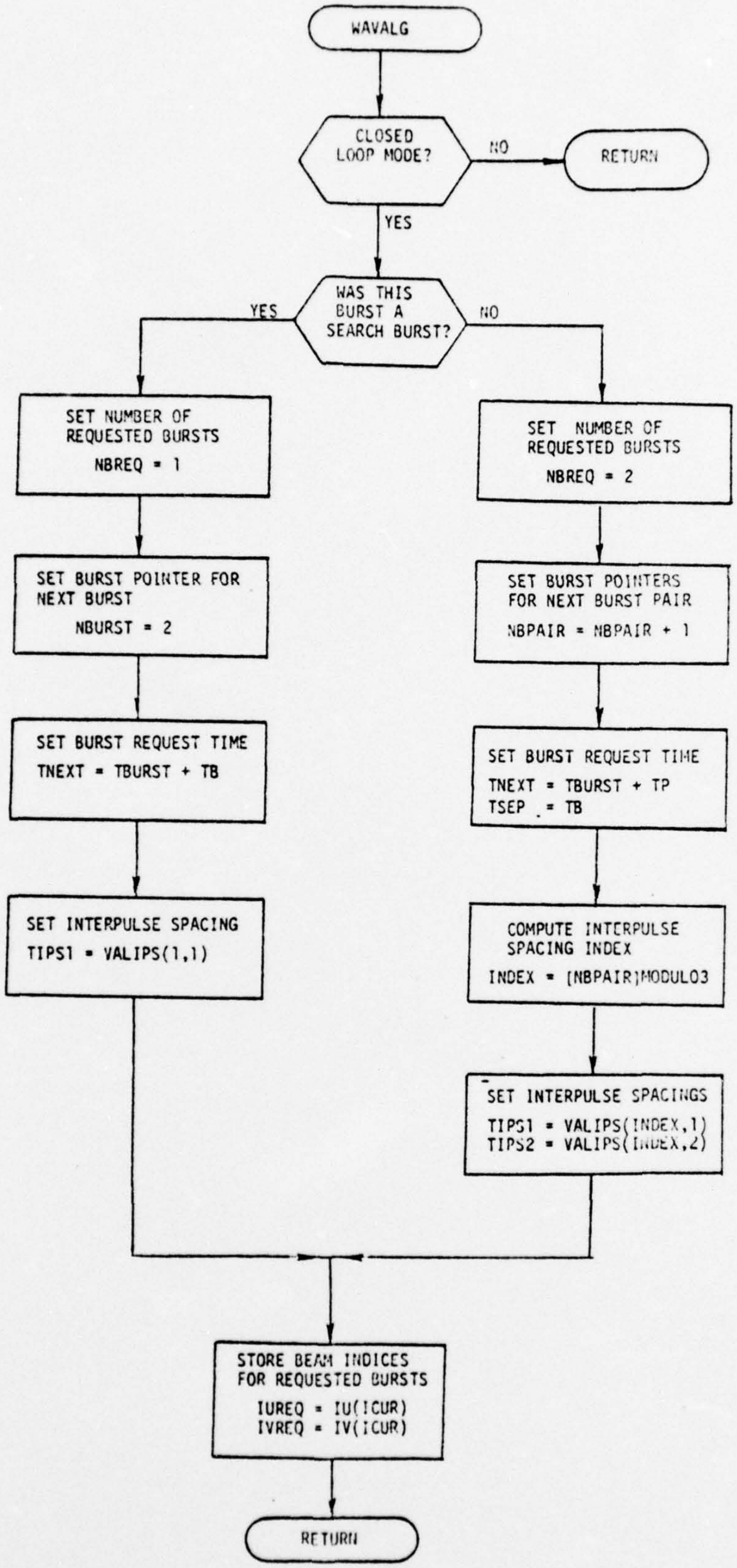


Figure 3-3 Waveform Request Algorithm Detailed Flowchart

4.0 PARAMETER SELECTION ALGORITHM

4.1 PURPOSE

The Parameter Selection Algorithm executes the real-time selection of ALF parameter values based on the measured environment. Since the development of an algorithm for choosing the N-M:N designation sequence has not been completed, the current Parameter Selection Algorithm specifies a predetermined N-M:N sequence.

Future versions will include an algorithm for selecting among a set of designation sequences.

4.2 GENERAL DESCRIPTION

Figure 4.1 is the overview flowchart for the Parameter Selection Algorithm (PARALG). The Parameter Selection Algorithm contains the predetermined values of M and N for the designation sequence. These values are passed to the CDG Algorithm for each beam via the CDG PARAMETERS file.

4.3 DATA SPECIFICATION

4.3.1 Input Data

The following data is input to the Parameter Selection Algorithm:

Program Symbol	Engineering Symbol	Description
IU	-	Index for beam U coordinate
IV	-	Index for beam V coordinate

4.3.2 Output Data

The following data is output to the CDG Algorithm via the CDG PARAMETERS file:

Program Symbol	Engineering Symbol	Description
NTRAN	N	Number of transmissions in designation sequence.
NMISS	M	Number of misses allowed in designation sequence.

4.3.3 Parameter Settings

Designation sequences with M = 1, N = 6; M = 2, N = 6; or M = 1, N = 5 are recommended.

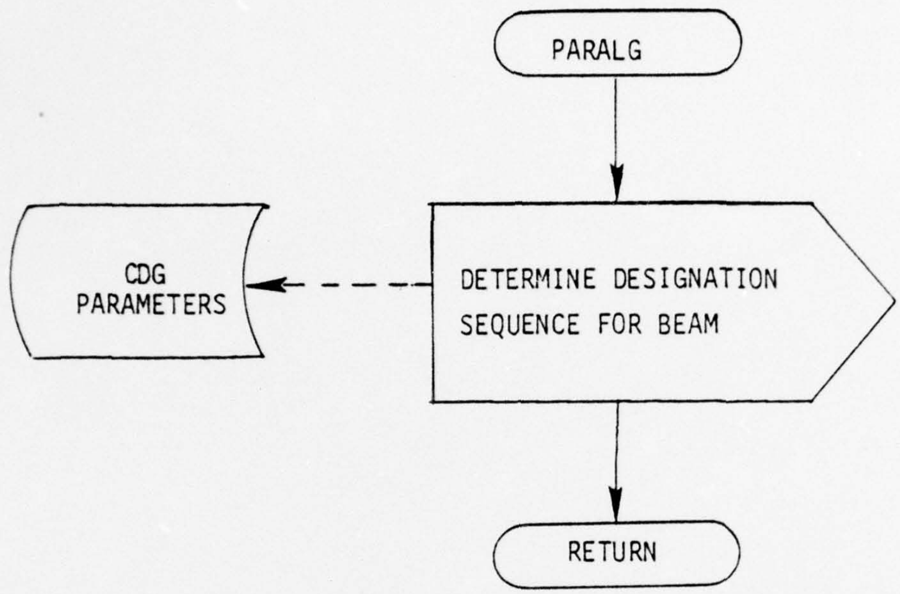


Figure 4.1 Parameter Selection Algorithm Overview Flowchart

5.0 MARKING ALGORITHM

5.1 PURPOSE

The Marking Algorithm operates on the sampled radar video from several Doppler filters to generate marks, i.e., estimates of the two-element state of range and range-rate for potential objects. Basically, the Marking Algorithm selects peaks in the radar video response (over a portion of the range, range-rate plane) which have amplitudes characteristic of potential RVs. For each of these peaks the algorithm estimates the location in the range, range-rate plane and saves it for further processing.

5.2 GENERAL DESCRIPTION

The range, range-rate Marking Algorithm described in this report is designed to operate on sampled video (sampling rate in range equal to twice the nominal bandwidth) from Doppler filters spaced at 305 m/s in range rate. The algorithm should operate satisfactorily with Doppler filter spacing as large as one-half of the Doppler resolution. These restrictions on sampling rates in range and Doppler were imposed to eliminate the requirement for sophisticated interpolation techniques.

Since the input data are assumed to be calibrated in range and range-rate, all operations are described in these units (rather than time and Doppler frequency). Video samples are provided for all Doppler filters within the RV range-rate acceptance region plus one filter on each side of the region. This allows marks to be generated in one filter beyond each edge of the RV range-rate acceptance region. The acceptance of marks in one filter beyond the RV range-rate acceptance region reduces the probability of missing an RV due to the Doppler resolution of the waveform.

A range, range-rate mark is generated by either of the following two criteria:

1. A peak in range-rate occurs at the same sample location as a peak in range, and the video amplitude at that sample exceeds the detection threshold (Type 1 mark);

or

2. A peak in range-rate occurs one range-rate sample away from a peak in range, the video amplitude at both samples exceeds the detection threshold, and a mark generated by the first criterion does not lie within plus or minus one sample spacing in range and plus or minus one sample spacing in range-rate (Type 2 mark).

The second criterion for generating a mark reduces leakage caused by interference from fragments or fragment ambiguities at lower range-rates. The final estimates of range and range-rate for each mark are computed by linear interpolation of the differences between sample amplitudes.

Figure 5.1 is an overview flow chart showing sequence of events and data flow for the marking algorithm. The range, range-rate marking algorithm first examines video data to locate peaks in range which exceed the detection threshold, and then proceeds to locate peaks in range-rate at each of these range locations. The resulting range, range-rate marks are tested for satisfaction of the marking criteria.

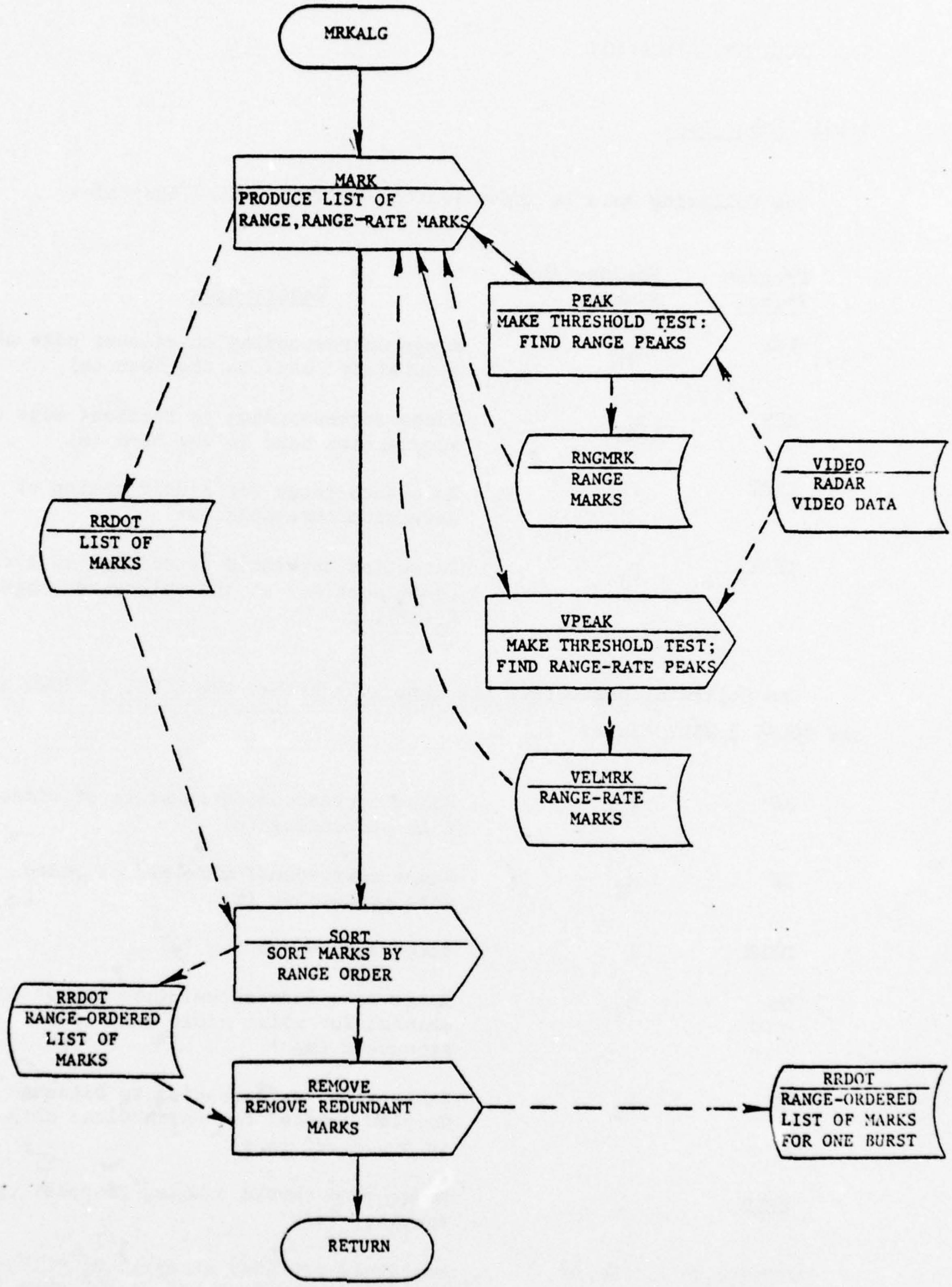


Figure 5.1 Marking Subsystem Overview Flowchart

5.3 DATA SPECIFICATION

5.3.1 Input Data

The following data is input via the MARKING PARAMETERS file:

Program Symbol	Engineering Symbol	Description
RSW	R_{SW}	Range corresponding to closest edge of acquisition band in the beam (m)
REW	R_{EW}	Range corresponding to furthest edge of acquisition band in the beam (m)
RREF	R_O	Reference range for specification of detection threshold (m)
THRESH	T_O	Detection threshold (specified as a radar cross section) at the reference range $R_O(m^2)$.

The following parameters are input via either the BURST 1 VIDEO or the BURST 2 VIDEO file:

RS	R_S	Range corresponding to start of video data processing (m)
RE	R_E	Range corresponding to end of video data processing (m)
DELR	Δ	Range sample spacing (m)
DS	D_S	Range-rate corresponding to lowest Doppler channel for which video data is processed (m/s)
DE	D_E	Range-rate corresponding to highest Doppler channel for which video data is processed (m/s)
DELD	ζ	Range-rate sample spacing (Doppler channel spacing) (m/s)
VIDEO(K,I)	$r(k;i)$	Amplitude of video at range $R_S + (k-1)\Delta$ and range-rate $D_S + (i-1)\zeta$. Note that $k = 1$ at R_S and $i = 1$ at D_S.

5.3.2 Output Data

The following data is output via either the BURST 1 MARKS or the BURST 2 MARKS file:

Program Symbol	Engineering Symbol	Description
NM	-	Number of range, range-rate marks in array RRDOT
RRDOT(Q,M)	$R\dot{R}(q,m)$	Range-ordered list, indexed by q, of range, range-rate marks generated for the current burst m = 1 Range (m) m = 2 Range-rate (m/sec) m = 3 Marking flag

5.3.3 Parameter Settings

The only parameters which must be specified in the range, range-rate marking algorithm are the detection threshold, T_O, and the reference range, R_O. Normally, the middle of the range corresponding to the acquisition band is used, i.e., $R_O = \dot{1}/2(R_{SW} + R_{EW})$. However, since the interference often cannot be treated as white gaussian noise, standard operating characteristics cannot be used exactly to achieve a desired P_D. A sensitivity analysis via simulation is the most reliable way to select the threshold. Values of detection threshold between $5*10^{-4}$ m^2 (-33 dBsm) and $3.16*10^{-4}$ m^2 (-35 dBsm) have been simulated.

5.4 MATHEMATICAL RELATIONSHIPS

Range marking is performed for one range sample at a time, starting with range R_S. At each range sample, range marking is performed for each of the Doppler channels. This is accomplished by the following sequence of operations. At range sample k, the successive amplitude differences (for Doppler channel i)

$$d(k;\ i) = r(k;\ i) - r(k-1;\ i) \quad ,$$

$$\frac{R_{SW} - R_S}{\Delta} + 2 \leq k \leq \frac{R_{EW} - R_S}{\Delta} + 1 \tag{5.1}$$

are computed. If $d(k;\ i) > 0$ and $d(k + 1;\ i) \leq 0$ and $r(k;\ i) \geq T_k$, a range mark is generated at range

$$R(j;\ i) = R_S + (k - \frac{3}{2})\Delta + \frac{d(k;i)}{d(k;\ i) - d(k+1;\ i)}\,\Delta \tag{5.2}$$

where j indicates the order, in range, of the mark for Doppler channel i. As indicated in equation (5.1), the marking is performed only over the range interval $[R_{SW}, R_{EW}]$. The detection threshold T_k, is varied with range according to

$$T_k = T_0^{1/2}\left(\frac{R_0}{R_S + (k - 1)\Delta}\right)^2 \tag{5.3}$$

Next, for each range mark $R(j;\ i)$, a check is made to determine if a peak in range-rate coincides with it. Each of the range marks is checked in order of range, regardless of Doppler channel. At the range sample nearest to $R(j;\ i)$, that is

$$k = \text{Integer}\left\{\frac{R(j;\ i) - R_S}{\Delta} + \frac{3}{2}\right\}, \tag{5.4}$$

the amplitude differences

$$e(k;\, i) = r(k;\, i) - r(k;\, i-1) \;,$$

$$i = 2, 3, \ldots, N_D \tag{5.5}$$

are computed, where

$$N_D = \frac{D_E - D_S}{\zeta} + 1$$

If $e(k;\, p) > 0$ and $e(k;\, p+1) \leq 0$ and $r(k;\, p) \geq T_k$, a range-rate mark is generated at range-rate

$$\dot{R}(k;\, \ell) = D_S + (p - \tfrac{3}{2})\zeta + \frac{e(k;\, p)}{e(k;\, p) - e(k;\, p+1)}\,\zeta \tag{5.6}$$

where ℓ indicates the order, in range-rate, of range-rate peaks at range sample k. If no peaks in range-rate are found, the range mark $R(j;\, i)$ is dropped. If one or more range-rate marks are generated at range sample k, each is tested to determine if it coincides with the range mark $R(j;\, i)$. If

$$|\dot{R}(k;\, \ell) - D_S - (i-1)\zeta| \leq \zeta \;, \tag{5.7}$$

then the range estimate $R(j;\, i)$ and the range-rate estimate $\dot{R}(k;\, \ell)$ are stored in positions $m = 1$ and $m = 2$ respectively of the array $R\dot{R}(q,\, m)$, where q indicates the order in which the range, range-rate marks are generated. Position $m = 3$ in $R\dot{R}(q,\, m)$ is set as follows

$$R\dot{R}(q,3) = \begin{cases} 1 \;, & \text{if } |\dot{R}(k;\, \ell) - D_S - (i-1)\zeta| \leq \frac{\zeta}{2} \,. \\ 0 \;, & \text{otherwise} \end{cases}$$

After the range, range-rate marks $R\dot{R}(q, m)$ are generated as described above, they are reordered by increasing range. Marks are then retained in $R\dot{R}(q, m)$ if one of the following two conditions is satisfied:

1. $R\dot{R}(q, 3) = 1$

   or

2. $R\dot{R}(q, 3) = 0$ and there exists no $R\dot{R}(u, n)$, $u \neq q$, such that

   $R\dot{R}(u, 3) = 1$

   and

   $|R\dot{R}(q, 1) - R\dot{R}(u, 1)| \leq \Delta$

   and

   $|R\dot{R}(q, 2) - R\dot{R}(u, 2)| \leq \zeta$

All other range, range-rate marks are deleted.

5.5 DETAILED DESCRIPTIONS

5.5.1 Marking Algorithm Executive Subroutine (MRKALG)

5.5.1.1 Purpose

The executive subroutine MRKALG controls the process of marking video data from one burst.

5.5.1.2 Detailed Description

The operation of the executive is very straightforward, as shown in the detailed flowchart of Figure 5.2. The sort operation simplifies the search for redundant marks.

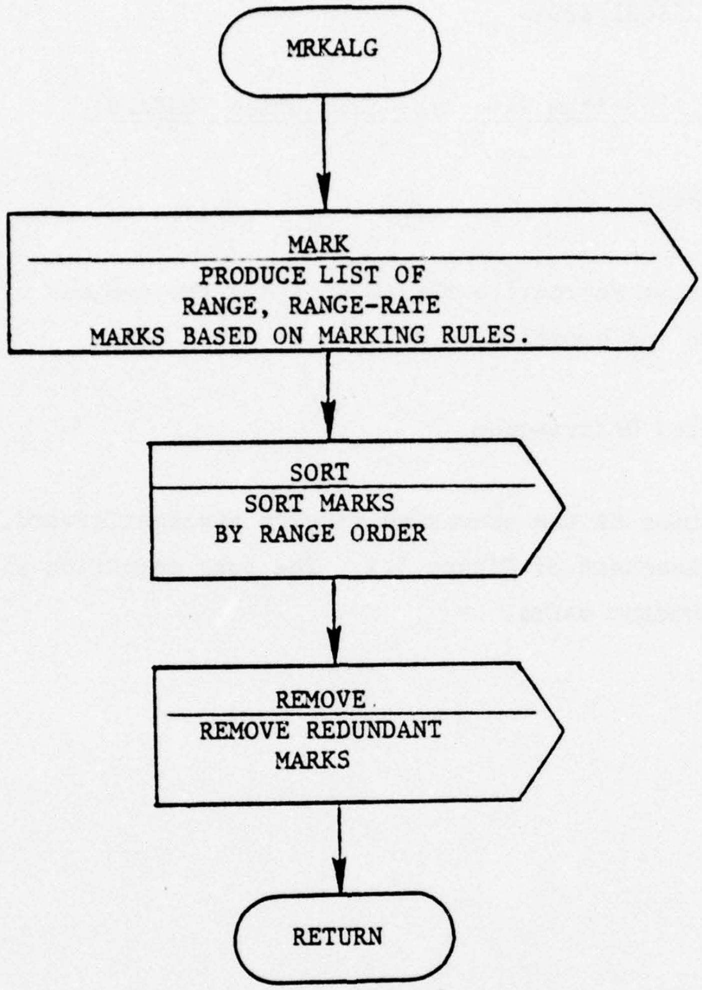


Figure 5.2 Detailed Flowchart for Marking Algorithm

5.5.2 MARK

5.5.2.1 Purpose

MARK controls the process of locating range and range-rate peaks. It then tests for satisfaction of the marking criterion (except for redundant mark test).

5.5.2.2 Input/Output Data

Input Data

Program Symbol	Engineering Symbol	Description
VIDEO(K,I)	$r(k,i)$	Amplitude of video at range $R_S + (K-1)\Delta$ and range-rate $D_S + (i-1)\zeta$

Output Data

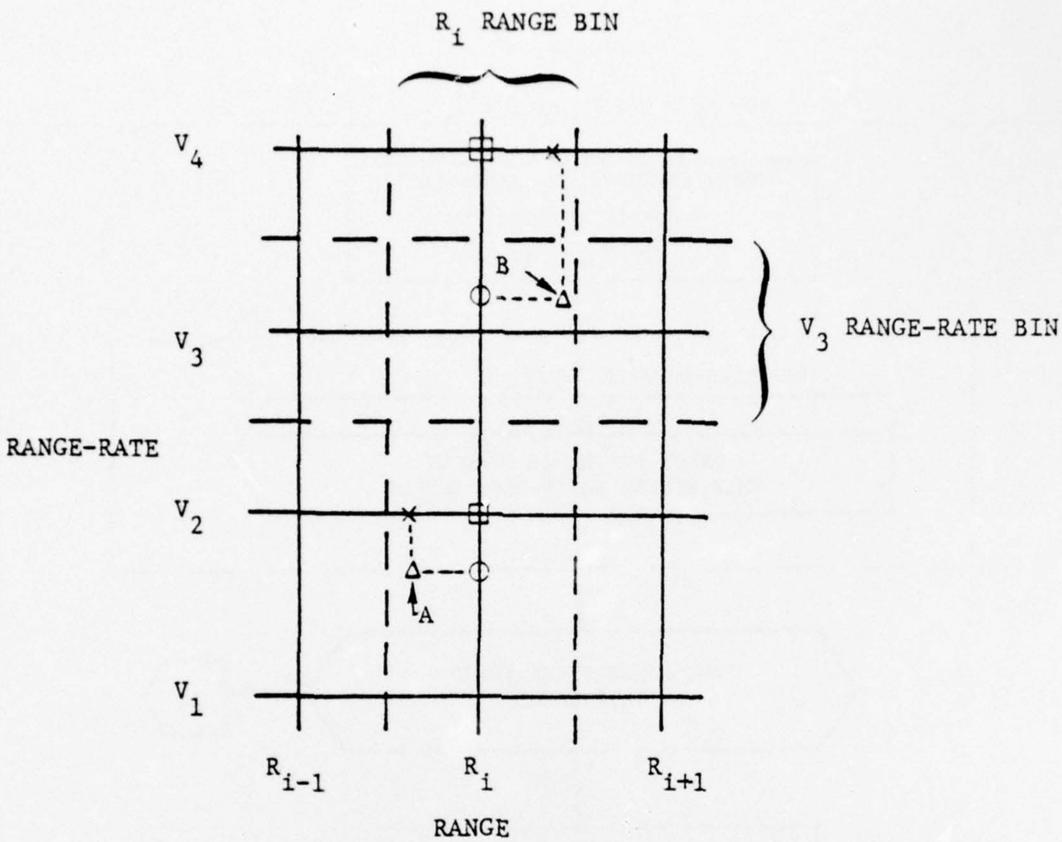
Program Symbol	Engineering Symbol	Description
RRDOT(Q,M)	$R\dot{R}(q,m)$	Range-ordered list of marks

5.5.2.3 Detailed Description

MARK produces range, range-rate marks by applying the marking rules specified in Section 5.4. The detailed flowchart of MARK is shown in Figure 5.4. The video data is examined one range bin at a time; subroutine PEAK is called to detect range peaks. Subroutine VPEAK is called to locate range-rate peaks. A search is then made to match each range-rate mark with the closest range mark. If both marks occur in the same range-rate bin (type 1 mark) a special flag is set for that range, range-rate mark. The only other marks retained are those where range and range-rate marks occur in adjacent range-rate bins (type 2 mark).

Figure 5.3 shows an example of the marking process for the range bin corresponding to range R_i. Range marks and r-nge-rate marks are shown for several samples where there are threshold crossings. The range, range-rate mark at A will be flagged as a type 1 mark; B is a type 2 mark.

The detailed flowchart of subroutine MARK is shown in Figure 5.4.



□ = Threshold Crossing

X = Range mark

O = Range-rate mark

Δ = Range, range-rate mark

Figure 5.3 Example of the Marking Process

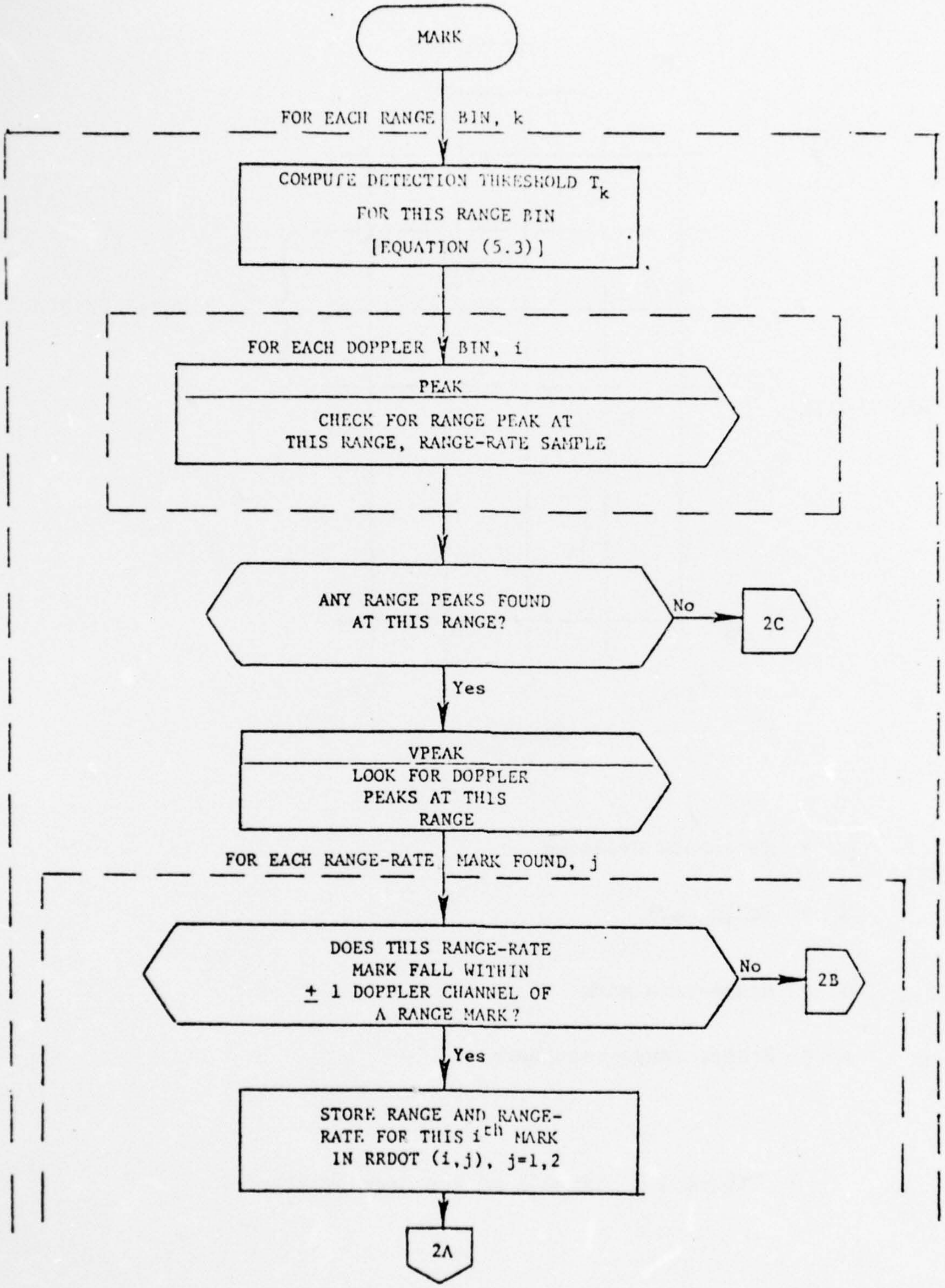


Figure 5.4 Subroutine MARK Detailed Flowchart

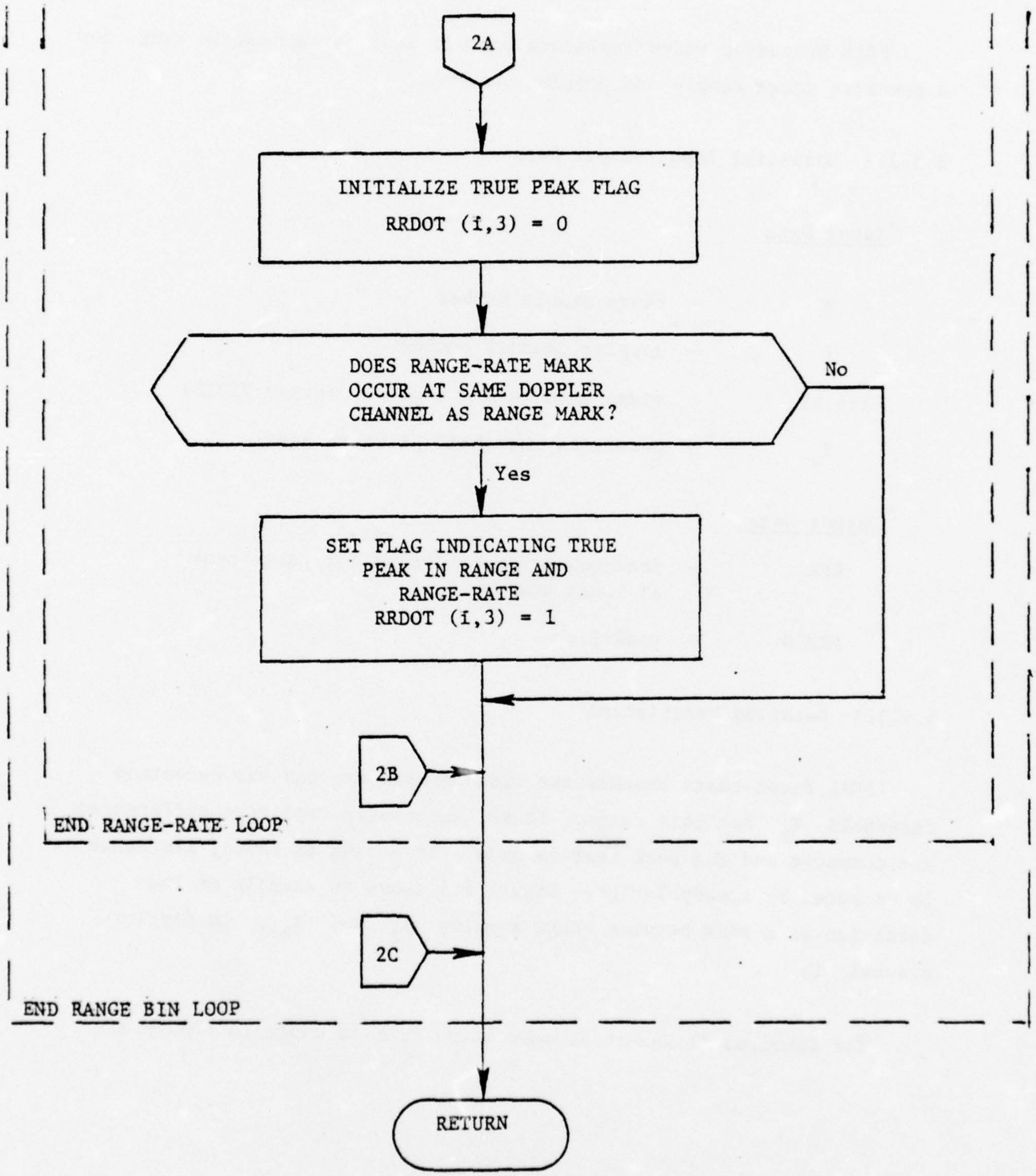


Figure 5.4 Subroutine MARK Detailed Flwochart (Cont'd)

5.5.3 PEAK

5.5.3.1 Purpose

PEAK processes video amplitude data to test for a peak in range for a specific range sample and doppler channel.

5.5.3.2 Essential Input/Output Data

Input Data

k	-	range sample number
i	-	doppler channel number
r(q,i)	-	video data ($k-1 \le q \le k+1$) (array VIDEO)
T_k	-	detection threshold at range sample k

Output Data

RPK	-	interpolated range value for range peak at range sample k
PKFLG	-	peak flag

5.5.3.3 Detailed Description

PEAK first tests whether the video sample exceeds the detection threshold T_k for this range. If so, successive amplitude differences are computed and the peak test is made. If a peak is found, its range is computed by interpolation. Figure 5.5 shows an example of the detection of a peak between range samples R_k and R_{k+1} in doppler channel i.

The detailed flowchart of subroutine PEAK is shown in Figure 5.6.

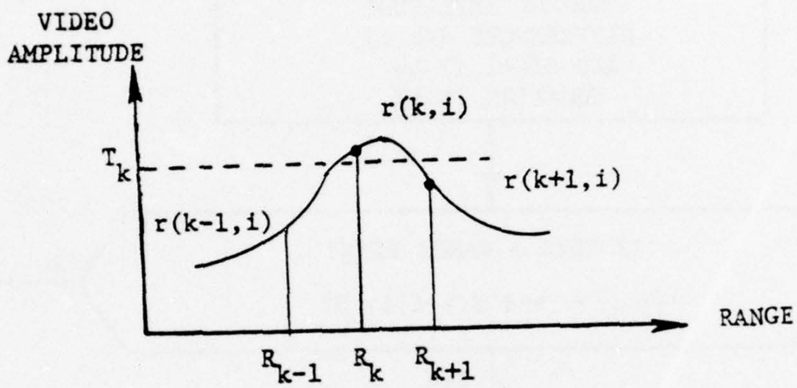


Figure 5.5 Peak Picking in Range

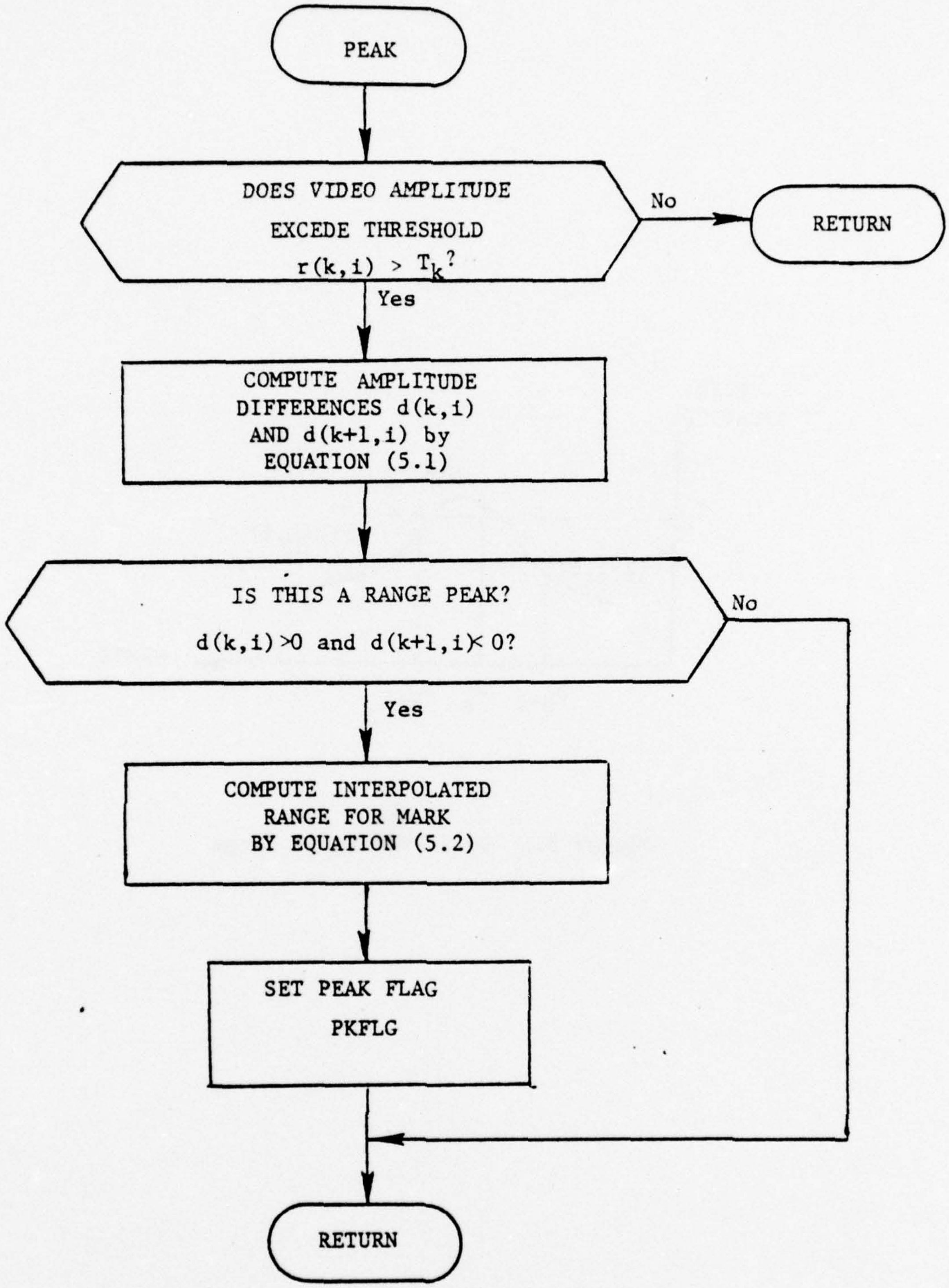


Figure 5.6 Subroutine PEAK Detailed Flowchart

5.5.4 VPEAK

5.5.4.1 Purpose

VPEAK processes video amplitude data to test for peaks in range-rate at a specified range sample.

5.5.4.2 Essential Input/Output Data

- Input Data

k	-	range sample number
r(k,i)	-	video (array VIDEO)

- Output Data

$\dot{R}(i)$	-	interpolated range-rates for range-rate peaks at this range sample (array VELMRK)
NPKS	-	number of peaks found

5.5.4.3 Detailed Description

Subroutine VPEAK performs essentially the same peak detecting functions in range-rate as are performed by PEAK for range. In addition, the interpolated range-rate is tested to determine whether it lies within the region of interest; if it lies outside the region, the mark is deleted.

The detailed flowchart of VPEAK is shown in Figure 5.7.

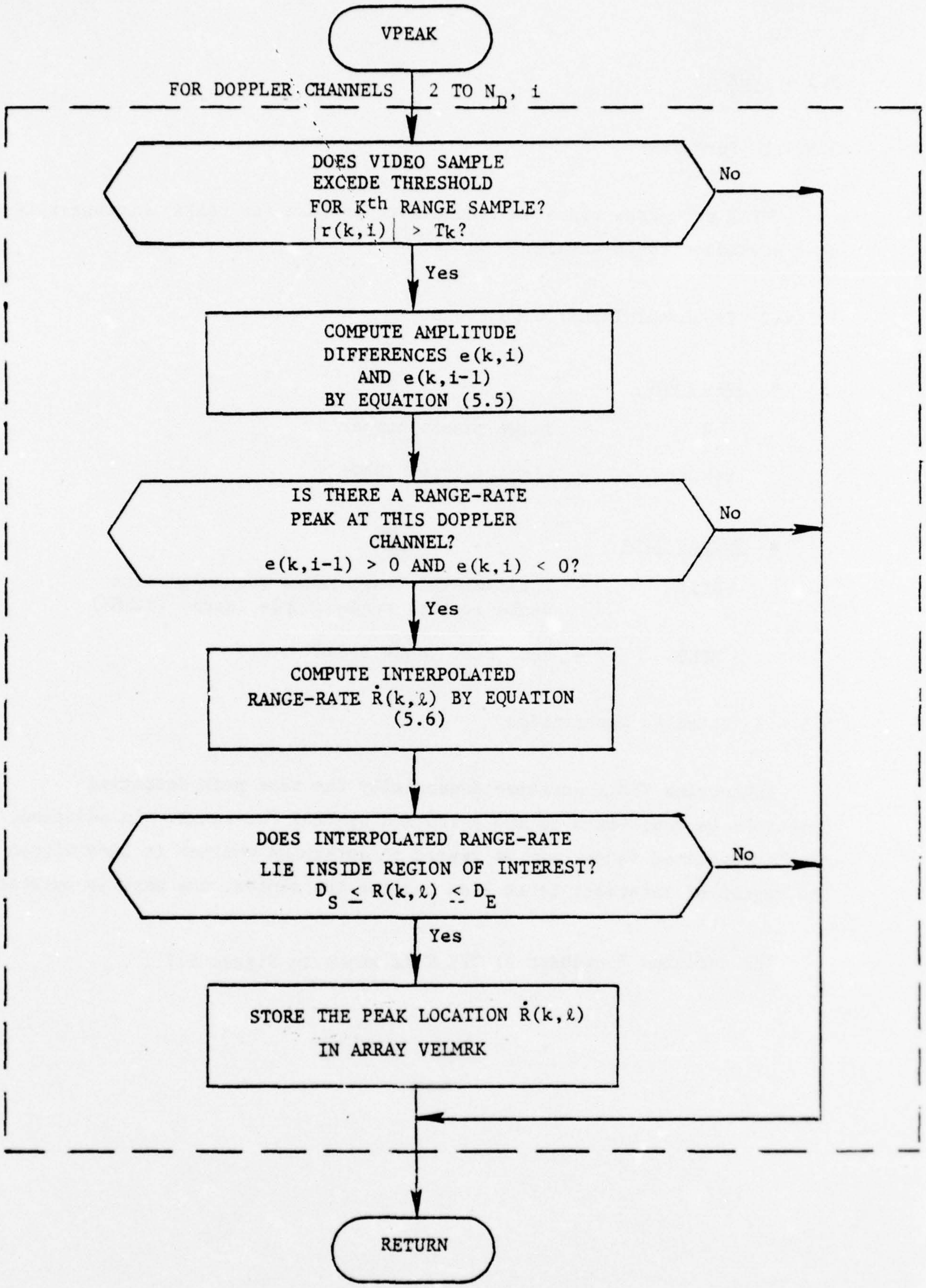


Figure 5.7 Subroutine VPEAK Detailed Flowchart

5.5.5 SORT

5.5.5.1 Purpose

The subroutine SORT sorts the $R\dot{R}$ marks on range (ascending order). The operation of this subroutine is self-explanatory.

5.5.6 REMOVE

5.5.6.1 Purpose

REMOVE searches the range-ordered list of marks and removes type 2 marks which are adjacent to type 1 marks, as described in the marking rules.

5.5.6.2 Data Flow

- Input Data

  $\dot{RR}(i,j)$ - range-ordered list of marks (array RRDOT)

- Output Data

  $\dot{RR}(i,j)$ - range-ordered list of marks (without redundant marks) (array RRDOT)

5.5.6.3 Detailed Description

REMOVE examines each type 1 mark (true peaks), searchs for and deletes any adjacent type 2 marks. Since the list is range-ordered, each search should involve only a few adjacent marks. Deleted marks are purged from the list.

The detailed flowchart of REMOVE is shown in Figure 5.8.

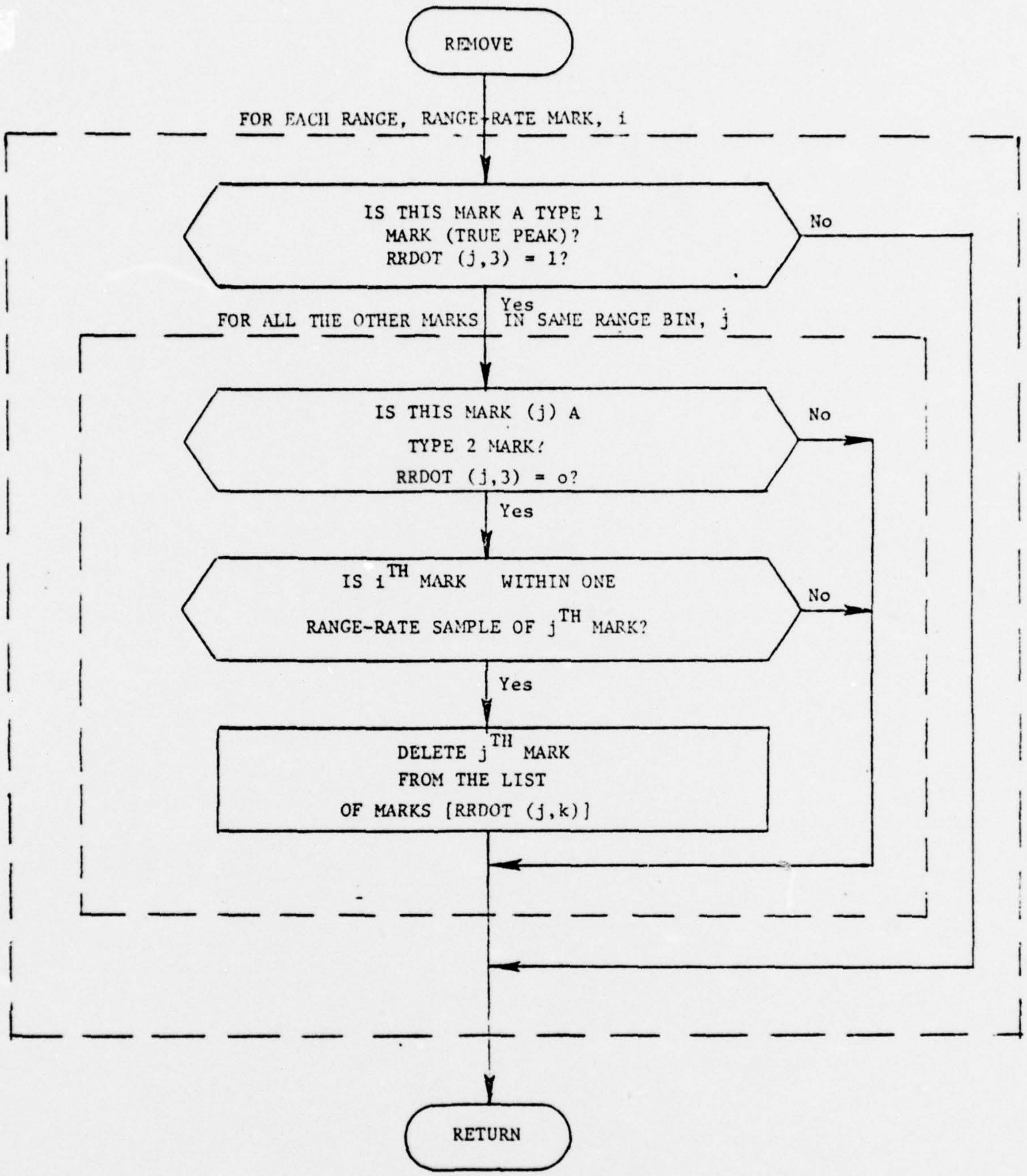


Figure 5.8 Subroutine REMOVE Detailed Flowchart

6.0 COINCIDENCE DETECTION ALGORITHM

6.1 PURPOSE OF SUBSYSTEM

The primary purpose of coincidence detection is the removal of marks caused by range ambiguities of the burst waveform.

Coincidence detection processing is required only for those transmissions on which a burst pair is employed.

6.2 GENERAL DESCRIPTION

The coincidence detection algorithm operates on the range, range-rate marks and the sampled video from two bursts on the same transmission to produce a set of marks for that transmission. In the bulk filtering application described here, a novel implementation of coincidence detection is used which has two equally important objectives

- Removing marks caused by range ambiguities of the burst waveform
- Achieving a high probability of RV detection in the presence of interference by fragments and fragment ambiguities.

This design has the advantage that in an environment with severe interference the algorithm sacrifices ambiguity removal performance to maintain an acceptable probability of detection, while in less severe interference the ambiguity removal performance improves substantially.

The coincidence detection algorithm recommended for the coherent bulk filter is designed to give equal importance to the removal of marks caused by range ambiguities and the minimization of RV leakage. To accomplish this, the algorithm departs from the conventional method of

implementing coincidence detection which requires that marks from the two bursts occur within a specified distance in range and range-rate. The recommended algorithm is defined as follows:

> A coincidence detection occurs at a given range and range-rate if (1) a range, range-rate mark is generated on either of the two bursts and, (2) the video from the other burst exceeds a threshold.

This algorithm reduces RV leakage in the important situation where a large fragment ambiguity is aligned in range with the RV return on one of the bursts. Since the fragment range-rate is lower than that of the RV, a mark is generated only at the lower range-rate. On the other burst, the ambiguity is shifted by virtue of the different pulse spacing, and a mark is generated at the RV range-rate. The recommended algorithm then generates a coincidence detection from the mark on the burst without interference and the video above the threshold on the burst without interference. The conventional implementation of coincidence detection would not generate a detection for this case.

An overview flowchart for the coincidence detection subsystem is shown in Figure 6.1. As shown in the overview flowchart, the coincidence detection algorithm first compares the marks from burst number 1 with the video from burst number 2, and then compares the marks from burst number 2 with the video from burst 1. The two lists of marks are then merged and sorted. Finally, redundant coincidence detections on the same object, resulting from marks on both bursts, are identified and removed.

The input data consists of marks from the marking algorithm and video data. The output consists of a list of marks passing coincidence detection. This list is passed to the designation sequence combining algorithm.

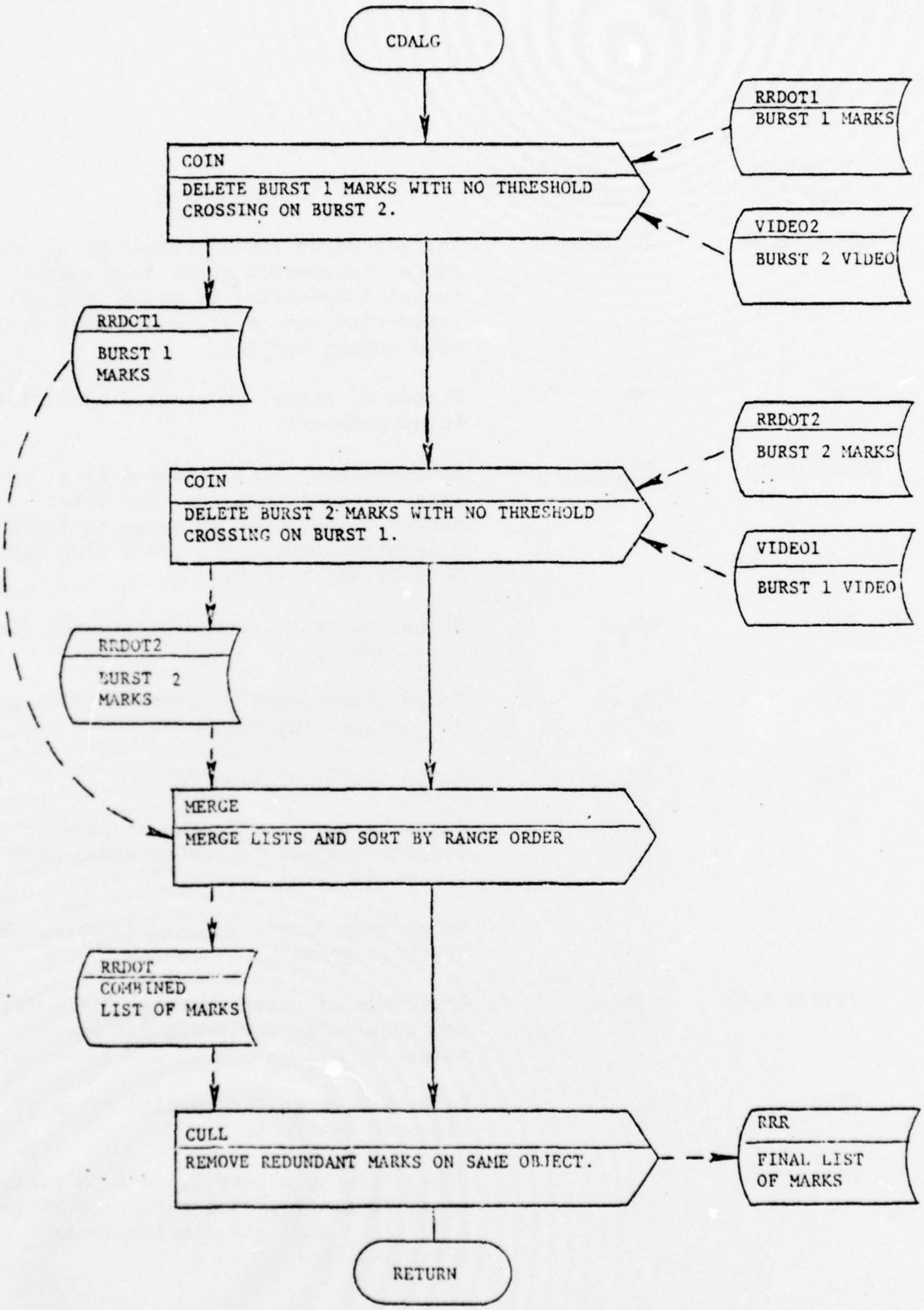


Figure 6.1 Overview Flowchart for Coincidence Detection Subsystem

6.3 DATA SPECIFICATION

6.3.1 Input Data

Program Symbol	Engineering Symbol	Description
RRDOT1(Q,M)	$R\dot{R}_1(q,m)$	Range-ordered list, indexed by q, of range, range-rate marks from burst number 1 specified by range (m=1,m), range-rate (m=2,m/s), and a flag (m=3) used during marking.
NM1	NM_1	Number of range, range-rate marks for burst number 1.
RRDOT2(Q,M)	$R\dot{R}_2(q,m)$	Range-ordered list, indexed by q, of range range-rate marks from burst number 2 specified by range (m=1,m), range-rate (m=2,m/s), and a flag (m=3) used during marking.
NM2	NM_2	Number of range, range-rate marks for burst number 2.
RS	R_S	Range corresponding to start of video data processing (m).
DELR	Δ	Range sample spacing (m).
DS	D_S	Range-rate corresponding to lowest Doppler channel for which video data is processed (m/s).
DELD		Range-rate sample spacing (Doppler channel spacing) (m/s).
VIDEO1(K,I)	$r_1(k,i)$	Amplitude of video at range $R_S+(k-1)\Delta$ and range-rate $D_S+(i+1)\zeta$ for burst number 2 (m).
RREF	R_0	Reference range for specification of video threshold (m).
THRCD	T_{CD}	Video threshold (specified as a radar cross section) at the reference range R_0 used to generate a coincidence detection (m^2).

6.3.1 (Continued)

Program Symbol	Engineering Symbol	Description
RCD	R_{CD}	Range interval used in the removal of redundant marks on the same object at the end of the coincidence detection operation (m).
RDOTCD	$\dot{R}_{CD}$	Range-rate interval used in the removal of redundant marks on the same object at the end of the coincidence detection operation (m/s).

6.3.2 Output Data

CDMRKS(L,M)	RRR(ℓ,m)	Range-ordered list, indexed by ℓ, of range, range-rate marks that have passed coincidence detection. m = 1 is range (m) m = 2 is range-rate (m/sec) m = 3 is angular coordinate, u m = 4 is angular coordinate, v m = 5 is the burst number.
NMC	NMC	Number of range, range-rate marks that have passed coincidence detection.

6.3.3 Parameter Settings

Four parameters can be specified in the coincidence detection algorithm. The reference range, R_O, at which the video threshold is specified is normally the middle of the range extent corresponding to the acquisition band, i.e., $R_O = 1/2\ (R_{SW} + R_{EW})$. The video threshold, T_{CD}, which is applied to the burst opposite that containing the mark, is set to achieve the desired operating point (P_D, P_{FA}) at the output of coincidence detection. Only cases with the video threshold, T_{CD}, set equal to the detection threshold, T_O, have been simulated. This was found to be a suitable operating point, and is recommended. The other two parameters, R_{CD} and $\dot{R}_{CD}$, the range and range-rate intervals, respectively, used in the removal of redundant marks on the same object, are selected based on the resolution properties of the waveform. It is recommended that they be set equal to twice the measurement covariances, i.e. $R_{CD} = 2\sigma_R$ and $\dot{R}_{CD} = 2\sigma_{\dot{R}}$.

6.4 MATHEMATICAL RELATIONSHIPS

A general flow diagram for the coincidence detection algorithm is shown in Figure 6.1.

The first step in the processing compares the range, range-rate marks from burst number 1 with the video from burst number 2. For each mark, $RR_1(\ell, m)$, from burst number 1, the indices of the closest video sample of burst number 2 are determined. The index in range is

$$k = \text{Integer} \left\{ \frac{R\dot{R}_2(\ell,1) - R_S}{\Delta} + \frac{3}{2} \right\} \tag{6.1}$$

and the index in range-rate is

$$i = \text{Integer} \left\{ \frac{R\dot{R}_2(\ell,2) - D_S}{\zeta} + \frac{3}{2} \right\} \tag{6.2}$$

The video amplitude of burst number 2 at range sample k and range-rate sample i is then compared with the video threshold, TC_k, given by

$$TC_k = T_{CD}^{1/2} \left[\frac{R_0}{R_S + (k-1)\Delta} \right]^2 \tag{6.3}$$

If

$$r_2(k,i) \geq TC_k \text{ ,}$$

the range range-rate mark $R\dot{R}_1(\ell,m)$ is retained; otherwise, it is deleted.

Similarly, the range, range-rate marks from burst number 2 are compared with the video from burst number 1. For each mark, $RR_2(\ell, m)$, the closest video sample from burst number 1 has range index

$$k = \text{Integer} \left\{ \frac{\dot{RR}_1(\ell,1) - R_S}{\Delta} + \frac{3}{2} \right\} \quad (6.4)$$

and range-rate index

$$i = \text{Integer} \left\{ \frac{\dot{RR}_1(\ell,2) - D_S}{\zeta} + \frac{3}{2} \right\} \quad (6.5)$$

If

$$r_1(k,i) \geq TC_k \quad ,$$

the mark $\dot{RR}_2(\ell, m)$ is retained; otherwise, it is deleted.

The range, range-rate marks retained from both bursts are merged into a range-ordered list, RRR (q,m), indexed on q, of marks specified by range (m = 1), range-rate (m = 2), and burst number (m = 3) on which the mark was generated. A test is then performed to identify redundant marks on the same object caused by the occurrence of marks on both bursts. If two marks $RRR(\ell,m)$ and $RRR(q,m)$ are found for which

$$RRR(\ell,3) \neq RRR(q,3) \quad (6.6)$$

and

$$|RRR(\ell,1) - RRR(q,1)| \leq R_{CD} \quad (6.7)$$

and

$$RRR(\ell,2) - RRR(q,2) \leq \dot{R}_{CD} \quad (6.8)$$

the mark with the minimum range is retained, and the other mark is deleted. (This is an arbitrary choice of which mark to retain.)

6.5 DETAILED DESCRIPTIONS

6.5.1 Executive Subroutine

6.5.1.1 Purpose

The coincidence detection executive subroutine controls the data flow and the execution of subroutines for this subsystem.

6.5.1.2 Detailed Description

The operation of the executive subroutine is straightforward. Subroutins are called as shown in the overview flowchart in Figure 6.1.

6.5.2 Subroutine COIN

6.5.2.1 Purpose

For each mark from a burst, COIN checks for a threshold crossing of the video from the other burst. If the video does not exceed the threshold, the mark is rejected.

6.5.2.2 Essential Input/Output Data

Input

$R\dot{R}_n(q,m)$	Range, range-rate marks from this burst (burst n)
$r_{n'}(k,i)$	Video from the other burst (burst n')

Output

$R\dot{R}_n(q,m)$	Range, range-rate marks from this burst pass first step of coincidence detection

6.5.2.3 Detailed Description

Subroutine COIN examines each mark from a burst. The range, range-rate sample closest to the mark is determined. The video of the other burst is examined at that sample to check for a threshold crossing. A detailed flowchart for COIN is shown in Figure 6.2.

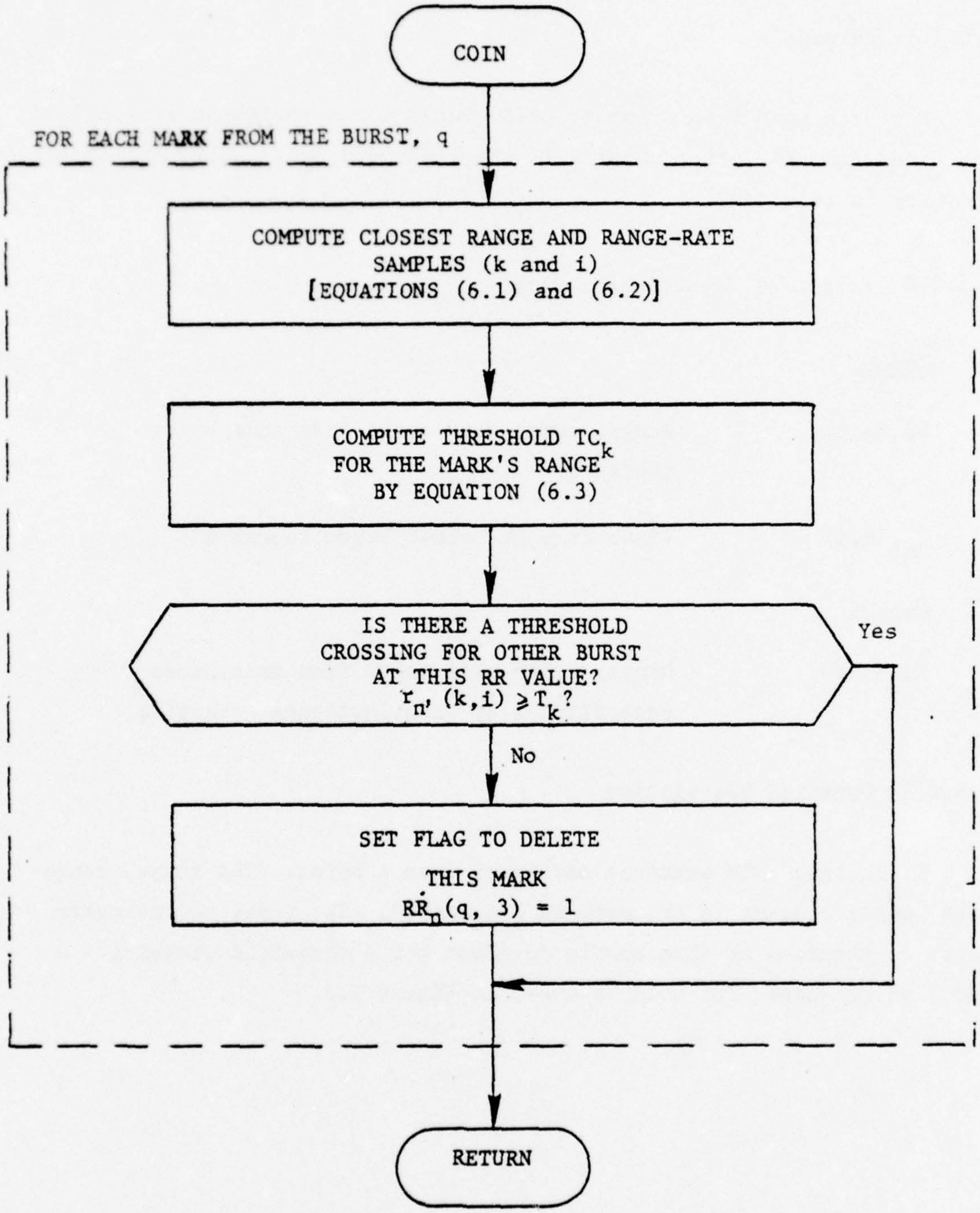


Figure 6.2 Subroutine COIN Detailed Flowchart

6.5.3 Subroutine MERGE

6.5.3.1 Purpose

MERGE combines the marks from the first and second bursts into one list, and then sorts the list in range order. Marks which have been flagged for deletion may be rejected at this point. The merge and sort operations are self explanatory.

6.5.4 Subroutine CULL

6.5.4.1 Purpose

CULL removes redundant marks caused by the same object.

6.5.4.2 Essential Input/Output Data

Input

RRDOT(q,m) - range-ordered list of marks from both bursts

Output

RRDOT(q,m) - range-ordered list of marks which passed test with redundant marks removed

6.5.4.3 Detailed Description

For each mark, CULL examines adjacent marks over a range extent of $\pm R_{CD}$. Only marks from the other burst are examined. If another mark is found whose range-rate differs from the current mark's by less than $\dot{R}_{CD}$, the mark with the lesser range is deleted.

A detailed flowchart for CULL is shown in Figure 6.3.

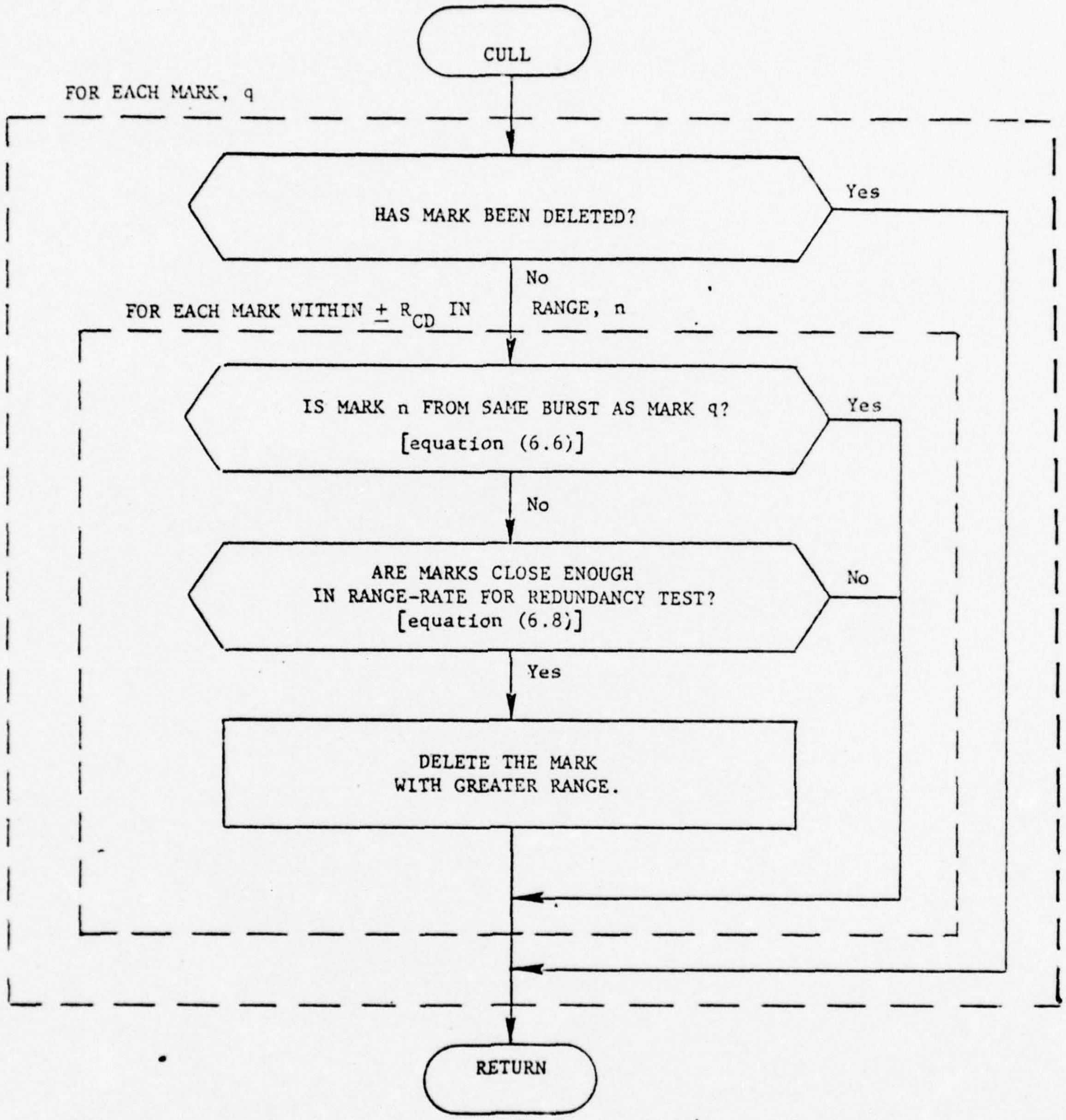


Figure 6.3 Subroutine CULL Detailed Flowchart

7.0 COHERENT DOUBLE GATING (CDG) ALGORITHM

7.1 PURPOSE OF SUBSYSTEM

The coherent double gating algorithm performs two general functions: (1) the removal of detections caused by clutter and ambiguities that are not removed by the single transmission processing, and (2) the generation of a state estimate on each threatening target for track initiation. The CDG algorithm performs the same processing after each transmission, regardless of whether it is a burst pair or a single burst.*

7.2 GENERAL DESCRIPTION

The CDG algorithm consists of three tests for reducing the number of potentially threatening objects designated in the beam and a maximum likelihood estimation computation for generating the best state estimate for track initiation.

The first test is a deceleration or gating test which eliminates any object whose trajectory deviates by some amount from a constant range-rate trajectory. To perform this test, the maximum likelihood estimate of the two-element state $(R,\dot{R})$ is computed for each object after transmission j-1. Based on this estimate and assuming a constant range-rate trajectory, an uncertainty region (elliptical gate) is computed in the range, range-rate plane for the measurement on transmission j. If the measurement (mark) falls within the gate, the object is retained. If the measurement falls outside of the gate, a miss is recorded. If the number of allowable misses is exceeded, the object is dropped from further consideration. If the number of allowable misses is not exceeded, the object's state estimate is updated to the present transmission time. Since a miss can be caused by failure to detect the object, it is preferable to allow at least one miss

*Although the CDG algorithm has this capability, this report specifies its use only with burst pairs.

during the sequence of transmissions to achieve a high probability of designation at the end of the sequence without requiring an unreasonably high single-hit probability of detection. A strategy allowing M misses in a total of N transmissions is denoted by N-M:N.

The CDG algorithm uses one of two methods to achieve an N-M:N strategy which yield different operating points. The type 1 N-M:N strategy, or N-M:N(1), requires that the object be detected on the first transmission, i.e., no new object state estimates will be initiated on transmissions 2 through N. The type 2 N-M:N strategy, or N-M:N(2), allows a miss to occur on the first transmission. This means that marks not falling into an elliptical gate on the second transmission are used to initialize new object state estimates with one miss. The type 2 strategy allows entries to be added to the CDG file on two transmissions, and hence achieves a higher probability of designation at the expense of more computations and a slightly higher level of ghosting.

The second test in the CDG algorithm is a velocity test. This test utilizes the increased confidence in the range-rate estimate from multiple transmissions to effectively sharpen the cut-off at the edges of the RV range-rate acceptance region. It is used after each transmission beyond the first and tests whether the range-rate estimate is, with high confidence, outside of the acceptance region.

The third test, called an association test, is designed to eliminate multiple entries in the CDG algorithm file corresponding to the same object. Such multiple entries can arise from multiple marks on the same object from the first transmission, or from multiple marks falling in the gate for the gating test on the second transmission. The association test is performed after each transmission beyond the second.

The maximum likelihood estimation of $(R,\dot{R})$ for track initiation is generated as part of the deceleration or gating test.

Figure 7.1 is an overview flowchart for the coherent double gating algorithm. During the first iteration shown in the flowchart (j=1) the list of object state estimates is initialized with the marks from transmission number one. These marks can be generated by the processing described either in Section 6 for burst pair transmissions or in Section 5 for single burst transmissions.

On the second iteration (j=2), each object state estimate from iteration one is used to establish a range, range-rate gate for the deceleration test. Marks from transmission two passing the gating test are used to update state estimates of the object. Those objects which then pass the velocity test are placed in the updated object file. Finally, if the strategy type is 2, marks not falling in any gate are used to initiate new object state estimates.

From the third iteration on, state estimates are updated by the mark closest to the center of the gate. Each object must then pass the velocity test. The remaining objects are processed by the association test to remove redundant tracks. After the last iteration, the list of objects is passed on to the Known Object Recognition Algorithm.

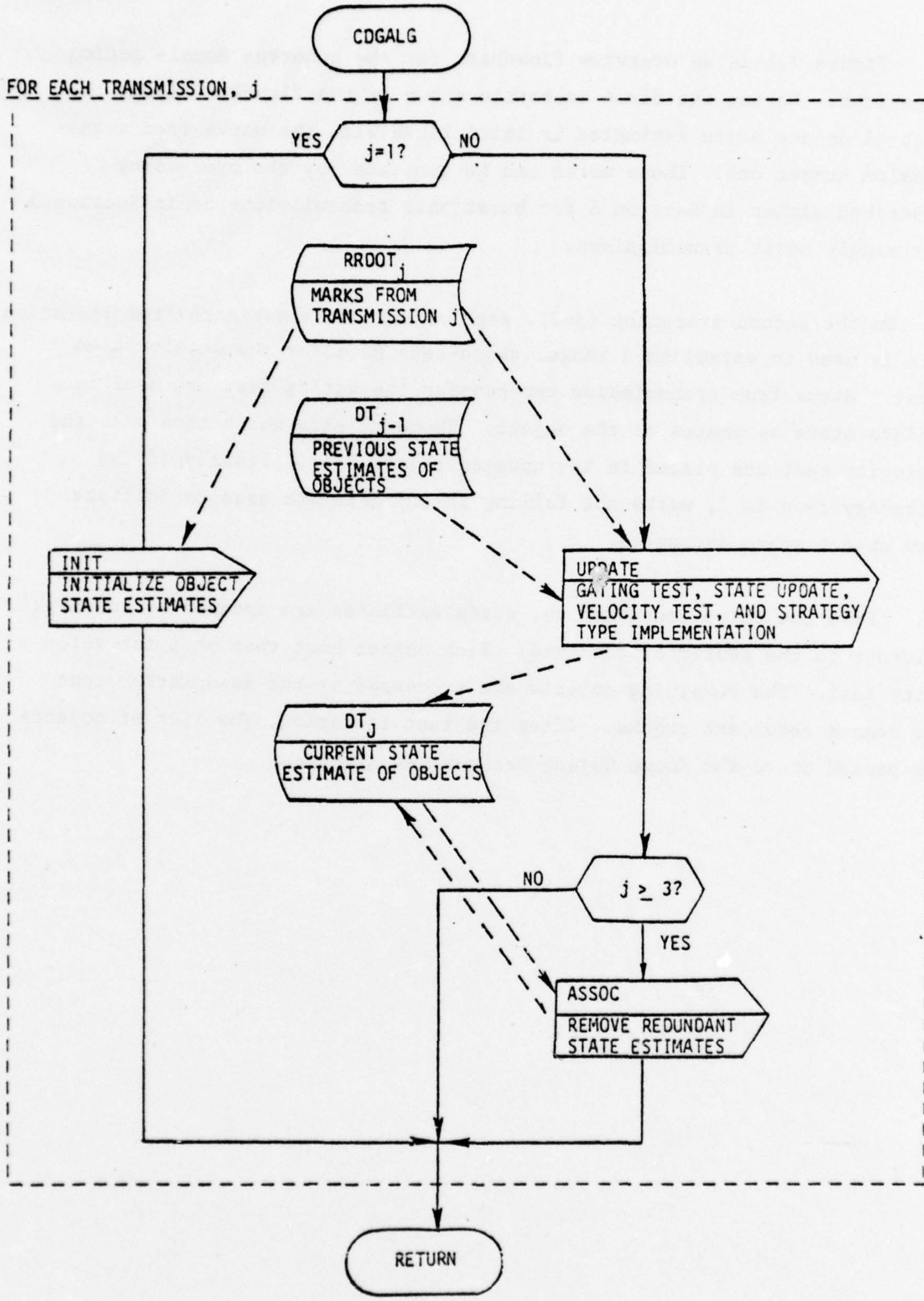


Figure 7.1 Overview Flowchart for Coherent Double Gating Algorithm Subsystem

7.3 DATA SPECIFICATION

7.3.1 Input Data

The following data is input via the CDG OPERATING PARAMETERS File:

Program Symbol	Engineering Symbol	Description
NTRAN	N	Number of transmissions in CDG sequence
NMISS	M	Number of misses allowed in CDG sequence
IS	IS	Specifies an N-M:N(IS) algorithm (IS=1 requires a detection on the first transmission, IS=2 requires a detection on first or second transmission).
CG	CG	Constant related to the probability of a detection from a threatening target falling in the elliptical gate.
CV	CV	Constant related to the probability of a threatening target passing the velocity test.
RDTMIN	$\dot{R}_{min}$	Lower range-rate limit in velocity test (m/sec).
RDTMAX	$\dot{R}_{max}$	Upper range-rate limit in velocity test (m/sec).
CA	CA	Constant related to the probability of incorrectly associating two detections from different objects.

The following data is input via the COINCIDENCE DETECTION MARKS File:

Program Symbol	Engineering Symbol	Description
CDMRKS(L,M)	$R\dot{R}_j(\ell,m)$	Range-ordered list, indexed by q, of range, range-rate marks from transmission j m = 1 range (m) m = 2 range-rate (m/sec) m = 3 angular coordinate, U m = 4 angular coordinate, V m = 5 burst number

NCD	NM_j	Number of range, range-rate marks in array CDMRKS
TMEAS	t_j	Time of transmission j (sec).

The following data is input via the CDG TRACK file:

NTN	NT_{j-1}	Number of objects passing the CDG Algorithm on transmission j-1 (number of entries in DT_{j-1}).
CDGTRK(K,M)	$DT_{j-1}(k,m)$	Range-ordered designation list, indexed by k, of objects remaining in the CDG algorithm after processing of transmission j.

M = 1	Range estimate (m)
M = 2	Range-rate estimate (m/s)
M = 3	Number of misses
M = 4	Variance of range estimate (m^2)
M = 5	Covariance of range and range-rate estimates (m^2/s)
M = 6	Variance of range-rate estimate (m^2/s^2)
M = 7	Angular coordinate U
M = 8	Angular coordinate V

The following data is input via the MEASUREMENT PARAMETERS file:

SIGR	σ	A-priori range error standard deviation (m)
SIGRDT	$\sigma_{\dot{r}}$	A-priori range-rate error standard deviation (m/sec)

7.3.2 Ouput Data

The following data is output via the CDG TRACK file:

NTN	NT_j	Number of objects passing the CDG algorithm on transmission j
CDGTRK (K,M)	$DT_j(k,m)$	Updated designation list of objects remaining after processing transmission j

7.3.3 Parameter Settings

The ten parameters N, M, IS, σ_r, $\sigma_{\dot{r}}$, CG, CV, $\dot{R}_{min}$, $\dot{R}_{max}$, and CA are used exclusively in the CDG Algorithm. $\dot{R}_{min}$ and $\dot{R}_{max}$ are set to the minimum and maximum range-rates of RV's threatening the defended area. Values of -7.2 km/s and -5.4 km/s respectively were used in all simulations. The constant CV is set to 2.57 to achieve a probability of .995 of a threatening target passing the velocity test. Simulation results with bursts of 16 10 MHz bandwidth pulses and coincidence detection indicate that settings of

N-M:N(IS)	= 5:6(2), 4:6(2), or 4:5(2)
$(\sigma_r, \sigma_{\dot{r}})$	= 3.66 m, 156 m/s
CG	= 10.6
CA	= 5.0 to 10.0

are suitable operating points. For 32 pulse bursts σ_r and $\sigma_{\dot{r}}$ can be reduced to approximately 1.83 m and 38.9 m/s.

7.4 MATHEMATICAL RELATIONS

A general flow diagram for the CDG algorithm is shown in Figure 7.1.

After the first (search) transmission, j=1, variables are initialized as follows.

$$ND_j = NM_j \qquad (7.1)$$

For each $q = 1,2,\ldots,NM_j$, set

$$\begin{aligned} DT_j(q,1) &= R\dot{R}_j(q,1) \\ DT_j(q,2) &= R\dot{R}_j(q,2) \\ DT_j(q,3) &= 0 \\ DT_j(q,4) &= \sigma_r^2 \\ DT_j(q,5) &= 0 \\ DT_j(q,6) &= \sigma_{\dot{r}}^2 \end{aligned} \qquad (7.2)$$

The remaining operations are performed after the second (first verify) and subsequent transmissions, j=2,3,... First, the measurement error covariance matrix is set to

$$R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \qquad (7.3)$$

the transition matrix is set to

$$\Phi = \begin{bmatrix} 1 & t_j - t_{j-1} \\ 0 & 1 \end{bmatrix} \qquad (7.4)$$

and the count of objects in DT_j is set to zero

$$ND_j = 0 \,. \tag{7.5}$$

For each object in the CDG algorithm after the processing of transmission j-1, that is for each $k = 1,2,\ldots,ND_{j-1}$, the following sequence of computations is performed.

The state estimate for object k after transmission j-1 is set to

$$\hat{X}_k = \begin{bmatrix} DT_{j-1}(k,1) \\ \\ DT_{j-1}(k,2) \end{bmatrix}$$

and the estimate covariance matrix for object k is set to

$$\hat{P}_k = \begin{bmatrix} DT_{j-1}(k,4) & DT_{j-1}(k,5) \\ \\ DT_{j-1}(k,5) & DT_{j-1}(k,6) \end{bmatrix}$$

The predicted state for object k at time t_j is

$$X_k = \Phi \hat{X}_k \tag{7.6}$$

and the predicted covariance for object k at time t_j is

$$P_k = \Phi \hat{P}_k \Phi^T \,. \tag{7.7}$$

The covariance matrix for the difference between the predicted state for object k and any one of the marks from transmission j is

$$V_k = P_k + R \tag{7.8}$$

For each mark q, $q=1,2,.,.,NM_j$, from transmission j, the difference between the predicted state for object k and the measured state of object q is

$$Z_{k,q} = X_k - \begin{bmatrix} \dot{RR}_j(q,1) \\ \dot{RR}_j(q,2) \end{bmatrix} \qquad (7.9)$$

The mark q falls within the acceptance gate of object k if

$$Z_{k,q}^T V_k^{-1} Z_{k,q} \leq CG \qquad (7.10)$$

At this point, two different paths are identified: Path I, if no marks fall within the acceptance gate; and Path II, if one or more marks fall within the acceptance gate.

Path I: No Marks in Acceptance Gate

If

$$DT_{j-1}(k,3) = M , \qquad (7.11)$$

where M is the allowed number of misses, delete object k from further tracking by simply beginning processing on object k+1. Object k will not appear in the updated BEAM DESIGNATION FILE in that case.

If

$$DT_{j-1}(k,3) < M ,$$

increment the designation counter

$$ND_j = ND_j + 1, \qquad (7.12)$$

increment the "miss" counter for object k

$$DT_j(ND_j,3) = DT_{j-1}(k,3) + 1 \qquad (7.13)$$

and set the other elements of the BEAM DESIGNATION FILE $DT_j(ND_j,m)$, equal to the predicted state and covariance of object k:

$$
\begin{aligned}
DT_j(ND_j,1) &= X_k(1) \\
DT_j(ND_j,2) &= X_k(2) \\
DT_j(ND_j,4) &= P_k(1,1) \\
DT_j(ND_j,5) &= P_k(1,2) \\
DT_j(ND_j,6) &= P_k(2,2)
\end{aligned}
\tag{7.14}
$$

Path II: Marks in Acceptance Gate

Within Path II, three cases can arise:

(1) If $j > 2$ and more than one detection falls within the acceptance gate of object k, only the mark q with the smallest value of $Z_{k,q}{}^T V_k^{-1} Z_{k,q}$ is retained and the others are deleted.

(2) If $j = 2$ and more than one mark falls within the acceptance gate, all such marks are retained for further processing. If $j=2$, the index, q of each mark retained is stored in the array, J. This array is required to implement an N-M:N(2) strategy.

(3) If one mark falls within the acceptance gate, it is retained for further processing.

For each mark retained by the gating test, the state estimate which results from the combination of object k and mark q

$$\hat{X}_q = X_k + K_k Z_{k,q} \tag{7.15}$$

where

$$K_k = P_k V_k^{-1}$$

and the estimate covariance matrix

$$\hat{P}_q = (I - K_k)P_k \qquad (7.16)$$

are updated, where I is the identity matrix. Each of these objects is processed through the <u>velocity test</u>:

$$\dot{R}_{max} + CV * \sqrt{\hat{P}_q(2,2)} \geq \hat{X}_q(2) \geq \dot{R}_{min} - CV * \sqrt{\hat{P}_q(2,2)} \qquad (7.17)$$

If the object passes this test, then the object is retained in the updated BEAM DESIGNATION FILE; otherwise, the object is deleted. For each object retained, set

$$ND_j = ND_j + 1 \qquad (7.18)$$

and

$$\begin{aligned}
DT_j(ND_j,1) &= \hat{X}_q(1)\\
DT_j(ND_j,2) &= \hat{X}_q(2)\\
DT_j(ND_j,3) &= DT_{j-1}(k,3)\\
DT_j(ND_j,4) &= \hat{P}_q(1,1)\\
DT_j(ND_j,5) &= \hat{P}_q(1,2)\\
DT_j(ND_j,6) &= \hat{P}_q(2,2) \qquad (7.19)
\end{aligned}$$

This is the end of the processing for each object, k, which remained in the CDG algorithm after transmission j-1.

If a type 2 strategy is used, the extra steps described in this paragraph must be carried out for j=2, i.e., after the second transmission. Each mark, not falling within an acceptance gate, i.e., $q \notin J$, is stored in the list of objects retained by the CDG Algorithm after transmission j. Each is treated as having been "missed" on the first transmission. For each such mark, q, set

$$ND_j = ND_j + 1 \tag{7.20}$$

and

$$\begin{aligned}
DT_j(ND_j,1) &= R\dot{R}_j(q,1) \\
DT_j(ND_j,2) &= R\dot{R}_j(q,2) \\
DT_j(ND_j,3) &= 1 \\
DT_j(ND_j,4) &= \sigma^2_R \\
DT_j(ND_j,5) &= 0 \\
DT_j(ND_j,6) &= \sigma^2_{\dot{R}}
\end{aligned} \tag{7.21}$$

When the list $DT_j(\ell,m)$ is completed, all entries are range-ordered.

For each transmission beyond the second, $j > 2$, an association test is performed on all objects retained in the CDG algorithm to eliminate redundant entries from the same object: two entries, ℓ and m, are said to be from the same object (associated) if

$$D^T B^{-1} D \leq CA \tag{7.22}$$

where

$$B = \begin{bmatrix} DT_j(\ell,4) & DT_j(\ell,5) \\ DT_j(\ell,5) & DT_j(m,6) \end{bmatrix} + \begin{bmatrix} DT_j(m,4) & DT_j(m,5) \\ DT_j(m,5) & DT_j(m,6) \end{bmatrix} \tag{7.23}$$

and

$$D = \begin{bmatrix} DT_j(\ell,1) - DT_j(m,1) \\ \\ DT_j(\ell,2) - DT_j(m,2) \end{bmatrix} \tag{7.24}$$

If entries ℓ and m are associated, the entry with the greater number of misses is deleted or, if $DT_j(\ell,3)$ and $DT_j(m,3)$ are equal, the entry with the greater range is deleted.

7.5 DETAILED DESCRIPTIONS

7.5.1 Executive Subroutine

7.5.1.1 Purpose

The coherent double gating algorithm executive subroutine controls data flow and execution of subroutines for this subsystem.

7.5.1.2 Detailed Description

The executive subroutine calls subroutines as shown in the overview flowchart, Figure 7.1.

7.5.2 Subroutine INIT

7.5.2.1 Purpose

On the first transmission, INIT performs the initialization described by equations (7.1) and (7.2). The array DT_1 is then ready for update on subsequent transmissions. A detailed flowchart of subroutine INIT is shown in Figure 7.2.

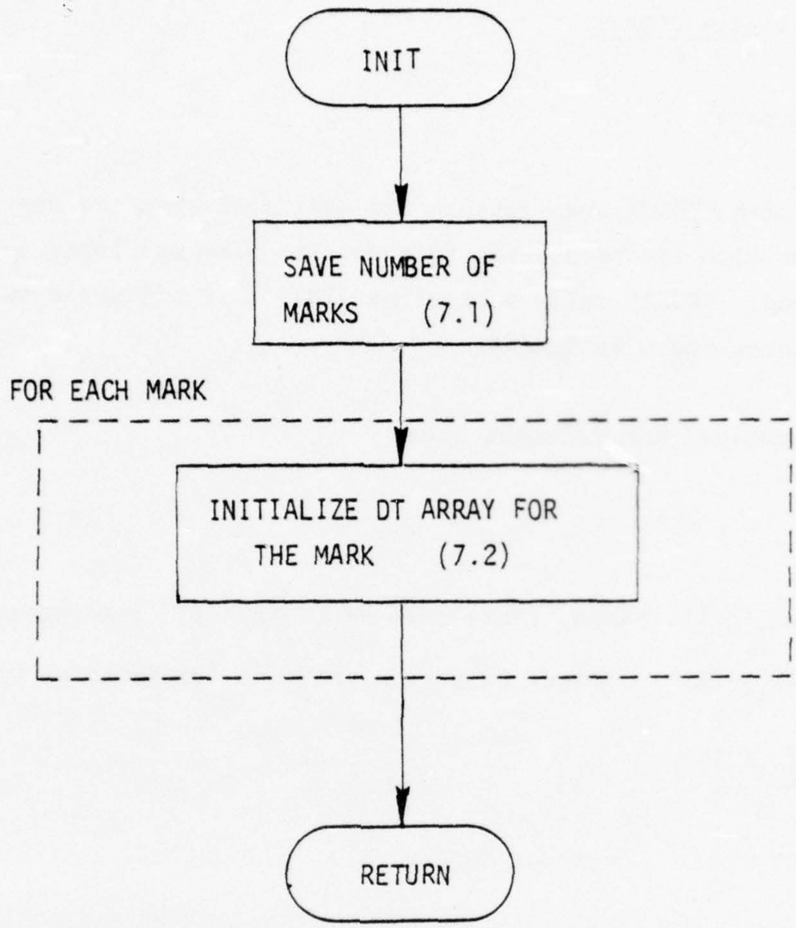


Figure 7.2 Subroutine INIT Detailed Flowchart

7.5.3 Subroutine UPDATE

7.5.3.1 Purpose

Subroutine UPDATE propagates state estimates from the previous transmission time to the present transmission time and implements the strategy type. UPDATE calls subroutine TESTS to perform the velocity test on updated state estimates.

7.5.3.2 Essential Input/Output Data

Input

$R\dot{R}_j(q,\ell)$ range, range-rate marks from j^{th} transmission

$DT_{j-1}(k,m)$ object file from previous transmission (j-1)

Output

$DT_j(k,m)$ current object file

7.5.3.3 Detailed Description

For each object in $DT_{j-1}(k,m)$, UPDATE performs the processing described by equations (7.3) through (7.14) and by equations (7.20) through (7.21). A detailed flowchart of UPDATE is shown in Figure 7.3.

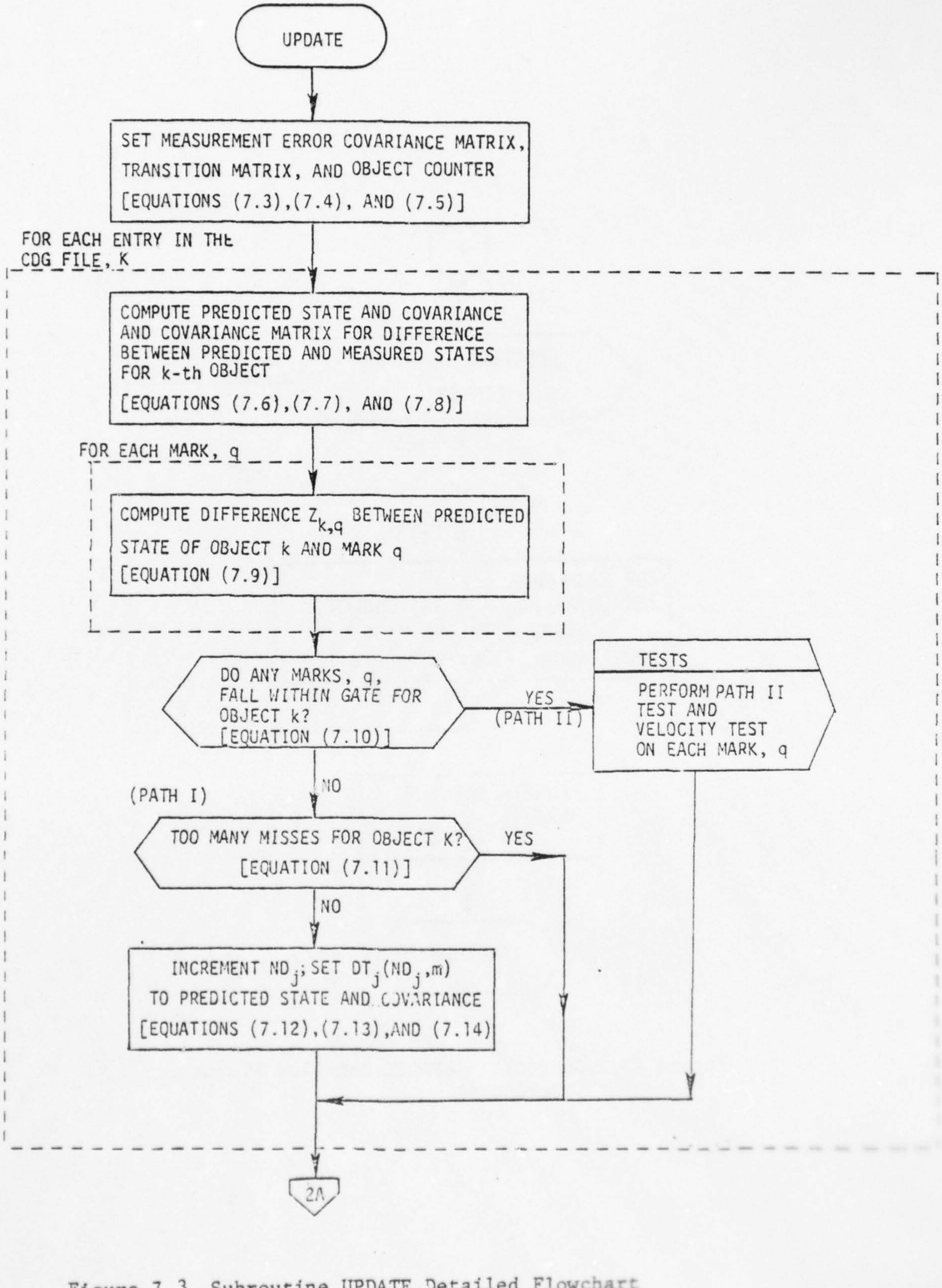


Figure 7.3 Subroutine UPDATE Detailed Flowchart

AD-A034 542 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 15/3.1
ACQUISITION LOGIC FUNCTION ALGORITHM SPECIFICATION.(U)
JUN 76 E R WENTZ, H MOSTAFAVI, H M PEARCE F19628-76-C-0002
UNCLASSIFIED ESD-TR-76-244 NL

2 OF 2
AD
A034542

END
DATE
FILMED
2-77

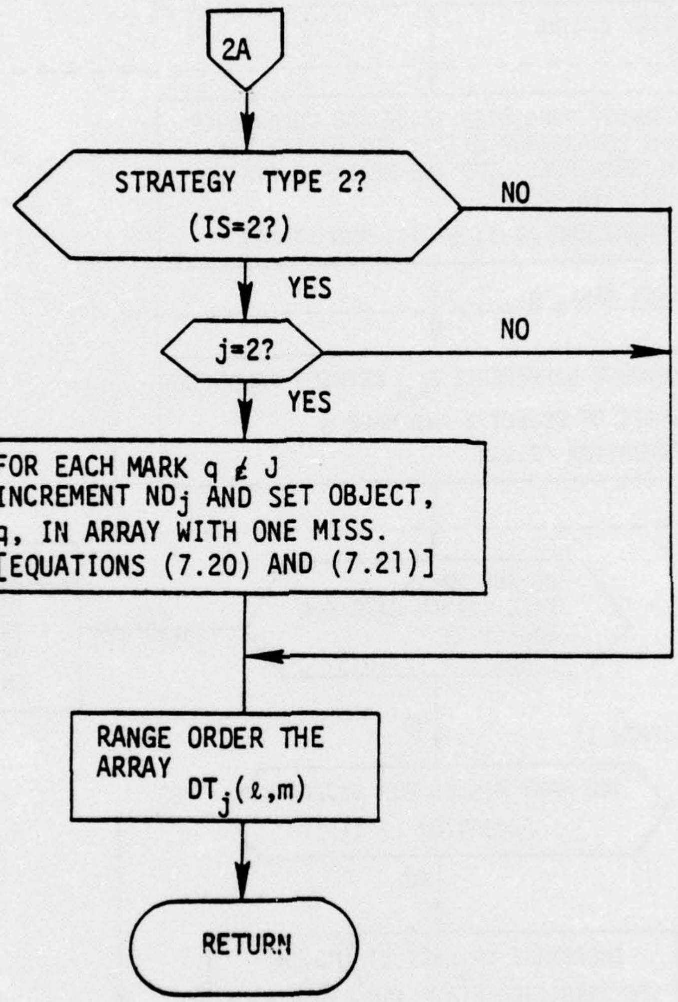


Figure 7.3 Subroutine UPDATE Detailed Flowchart (Continued)

7.5.4 Subroutine TESTS

7.5.4.1 Purpose

Subroutine TESTS performs two types of tests on the marks which fall within the acceptance region of an object's updated state estimate. These tests involve the choice of objects to be retained, and the velocity test.

7.5.4.2 Input/Output Data

Input

Program Symbol	Engineering Symbol	Description
QMARKS(I)	-	List of the indices, q, of each mark in RRDOT array which falls in acceptance region of object k
ZQK(I)	$Z_{q,k}$	List of differences between predicted and measured values for each mark in array QMARKS(I)
K	k	Index in $DT_{j-1}(k,m)$ object file for current object
VKINV(2,2)	V^{-1}_{k}	Inverse error covariance matrix for object k
XK(2)	$\bar{X}_k$	Predicted state for object k
PK(2,2)	P_k	Predicted covariance for object k
RDTMIN	$\dot{R}_{min}$	Lower range-rate limit for velocity test
RDTMAX	$\dot{R}_{max}$	Upper range-rate limit for velocity test
CV	CV	Constant related to the probability of a threatening target passing the velocity test

7.5.4.2 (Continued)

Output

Program Symbol	Engineering Symbol	Description
NDJ	ND_j	Current number of objects in updated BEAM DESIGNATION FILE
DESVEC(I,NDJ)	$DT_j(I,NDJ)$	Current list of objects in the updated BEAM DESIGNATION FILE
NJ2	-	Number of objects in array J2
J2(I)	J	List of indices q of each mark in the RRDOT array which falls in the acceptance region of an object being tracked

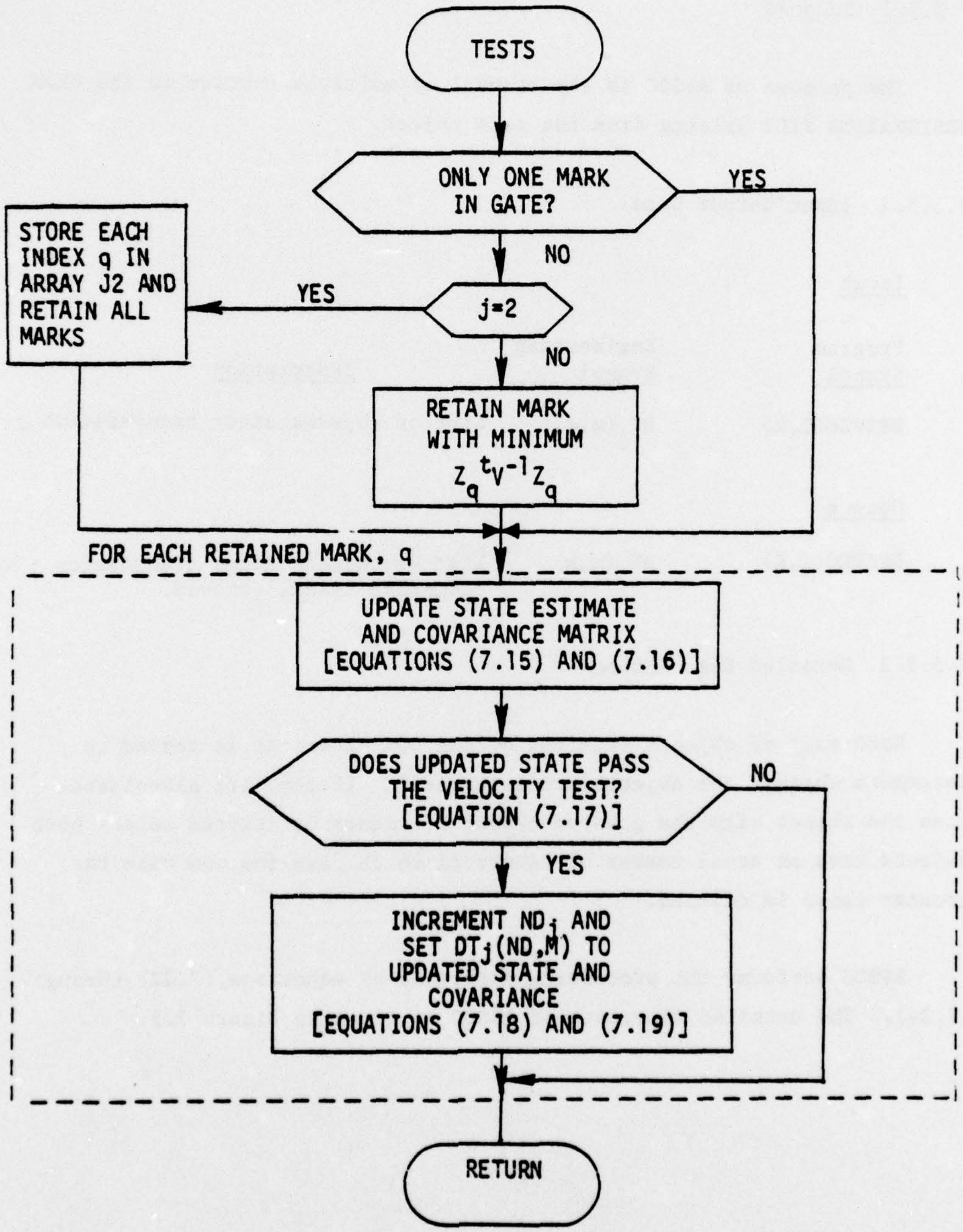


Figure 7.4 Subroutine TESTS Detailed Flowchart

7.5.5 Subroutine ASSOC

7.5.5.1 Purpose

The purpose of ASSOC is the removal of multiple entries in the BEAM DESIGNATION FILE arising from the same object.

7.5.5.2 Input/Output Data

Input

Program Symbol	Engineering Symbol	Description
DESVEC(I,K)	$DT_j(m,k)$	List of objects after transmission j.

Output

Program Symbol	Engineering Symbol	Description
DESVEC(I,K)	$DT_j(m,k)$	List of objects after transmision j with redundant tracks removed.

7.5.5.3 Detailed Description

Each pair of objects retained by the CDG Algorithm is tested to determine whether the objects are associated. If they are associated, then the object with the greater number of misses is deleted unless both objects have an equal number of misses in which case the one with the greater range is deleted.

ASSOC performs the processing described by equations (7.22) through (7.24). The detailed flowchart of ASSOC is given in Figure 7.5.

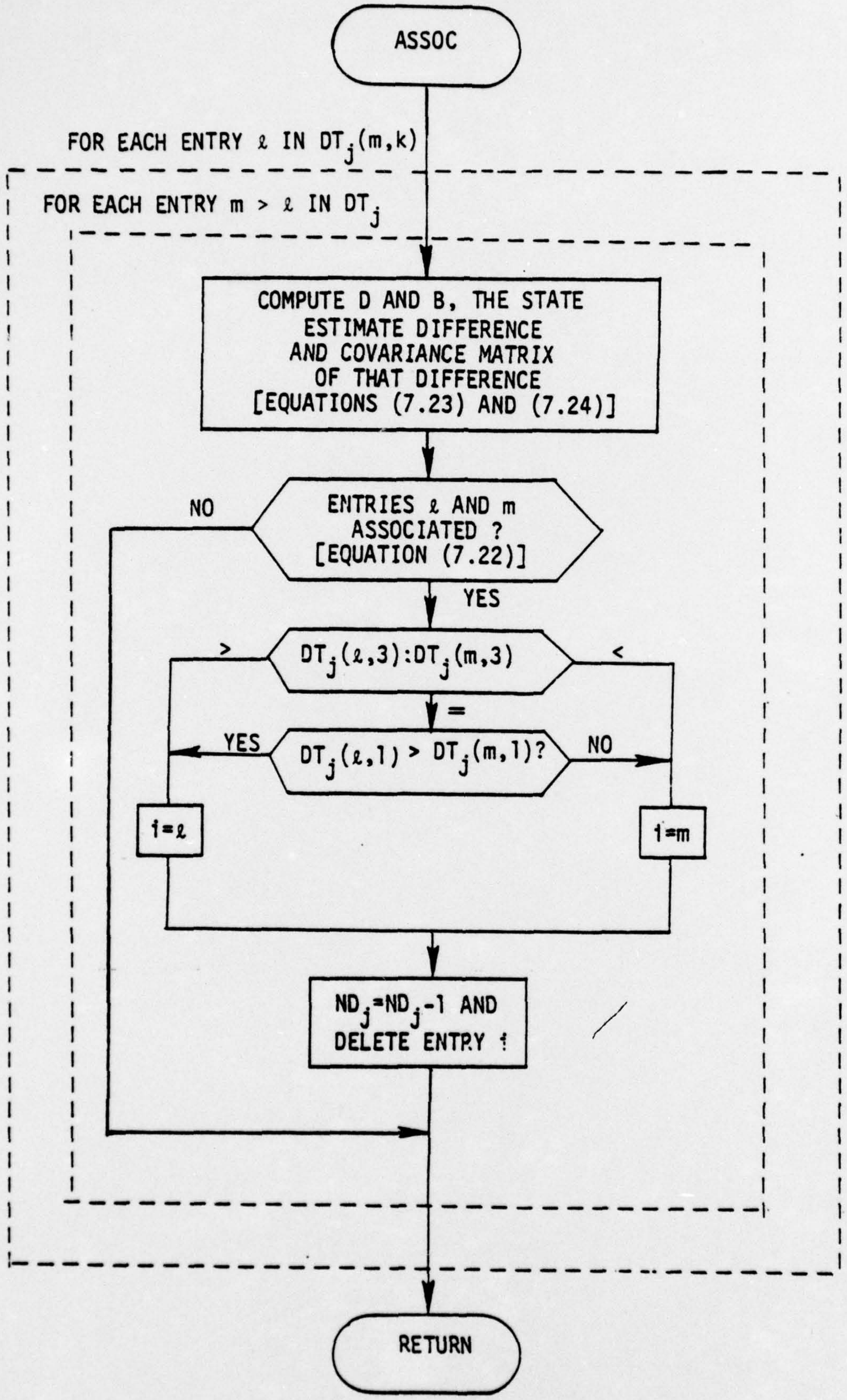


Figure 7.5 Subroutine ASSOC Detailed Flowchart

8.0 OBJECT BEAM POSITION ALGORITHM

8.1 PURPOSE OF SUBSYSTEM

The object beam position algorithm computes the UV coordinates for designated objects and stores those coordinates in the BEAM DESIGNATION FILE.

8.2 GENERAL DESCRIPTION

Two channels of monopulse data are available which provide object angular position (ΔU and ΔV) with respect to the beam center.

The object beam position algorithm obtains object angular measurements from the monopulse data for each object that passes coincidence detection. These monopulse measurements are combined with the beam UV coordinates to obtain object UV coordinates.

The overview flowchart for the Object Beam Position Algorithm is shown in Figure 8.1.

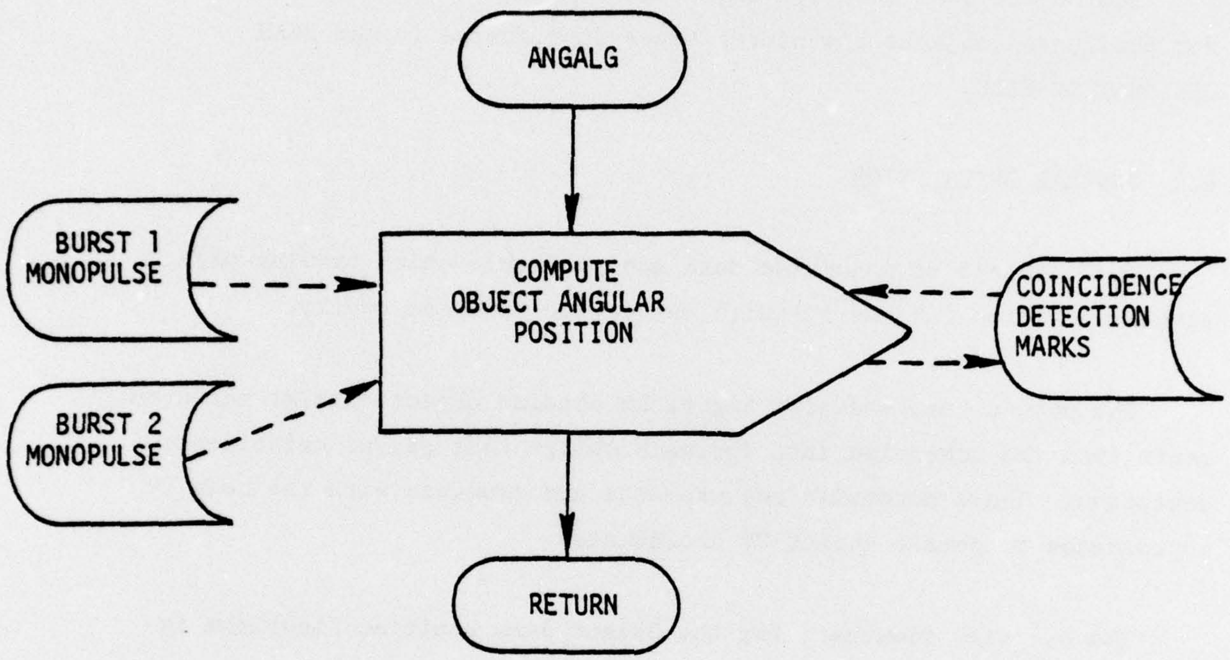


Figure 8.1 Object Beam Position Algorithm Overview Flowchart

8.3 DATA FLOW

8.3.1 Input Data

The following data is input from the ALF via the Beam Status file:

Program Symbol	Engineering Symbol	Description
IU	-	Beam U index
IV	-	Beam V index

The following data is input from the Coincidence Detection Algorithm via the COINCIDENCE DETECTION MARKS file.

NCD	-	Number of coincidence detection marks from the current burst pair
CDMRKS(L,M)	RRR(ℓ,m)	Range, range-rate marks from coincidence detection
	R	m = 1 is range (m)
	$\dot{R}$	m = 2 is range-rate (m/sec)
	i_b	m = 5 is the burst number
	-	ℓ is the mark index

The following data is input from the Matched Filter Processing Function via the BURST 1 MONOPULSE file and the BURST 2 MONOPULSE file:

RSM	R_S	Range corresponding to start of video data processing (m)
REM	R_E	Range corresponding to end of video data processing (m)
DEL	Δ	Range sample spacing (m)
DSM	D_S	Range-rate corresponding to lowest Doppler channel for which video data is processed (m/s)

8.3.1 (Continued)

DEM	D_E	Range-rate corresponding to highest Doppler channel for which video data is processed (m/s)
DELD		Range-rate sample spacing (Doppler channel spacing) (m/s)
UBST1(K,L)	U_{b_I}	Monopulse U measurement, beam 1
VBST1(K,L)	V_{b_1}	Monopulse V measurement, beam 1
UBST2(K,L)	U_{b_2}	Monopulse U measurement, beam 2
VBST2(K,L)	V_{b_2}	Monopulse V measurement, beam 2

For the four monopulse arrays, note that K = 1 at R_s and L = 1 at D_s.

The following data is input via the OBJECT BEAM POSITION PARAMETERS file:

UBEAM(IU,IV)	U_o	U coordinate (sine space) of the IU, IV beam
VBEAM(IU,IV)	V_o	V coordinate (sine space) of the IU,IV beam

8.3.2 Output Data

The following data is output via the COINCIDENCE DETECTION MARKS file:

CDMRKS(L,M)	-	This is the same file listed under Input Data, with angle data added to it.
	U	I = 3 is the U coordinate
	V	I = 4 is the V coordinate

8.4 MATHEMATICAL RELATIONSHIPS

Measurements of angular position are obtained from monopulse data for each mark which passes coincidence detection. These marks are contained in the input array CDMARKS.

For each mark the range index k of the closest range sample is computed from the range mark R

$$K = \text{Integer}\left\{\frac{R - R_s}{\Delta}\right\} \tag{8.1}$$

where Integer {j} means the integer closest to j.

For each mark, the range-rate index ℓ of the closest range-rate sample is computed from range-rate mark $\dot{R}$

$$\ell = \text{Integer}\left\{\frac{\dot{R} - D_s}{\zeta}\right\} \tag{8.2}$$

Each mark in the CDMARKS array contains a pointer, i_b, indicating which burst (1 or 2) produced the mark. The object's U and V positions within the beam is then computed as

$$\Delta U = \begin{cases} U_{b_1}(k,\ell) & , \quad i_b = 1 \\ U_{b_2}(k,\ell) & , \quad i_b = 2 \end{cases} \tag{8.3}$$

$$\Delta V = \begin{cases} V_{b_1}(k,\ell) & , \quad i_b = 1 \\ V_{b_2}(k,\ell) & , \quad i_b = 2 \end{cases} \tag{8.4}$$

The U and V coordinates of the beam center, U_o and V_o, are obtained from a prestored array, based on the beam indices, IU and IV.

The object U and V coordinates are then computed as

$$U = U_o + \Delta U \tag{8.5}$$

$$V = V_o + \Delta V \tag{8.6}$$

8.5 DETAILED DESCRIPTIONS

8.5.1 Object Beam Position Algorithm Executive Subroutine (ANGALG)

8.5.1.1 Purpose

The executive subroutine ANGALG controls the process of computing object angular positions.

8.5.1.2 Detailed Description

The detailed flow chart of the executive subroutine is shown in Figure 8.2. The computations follow the sequence described in Section 8.4.

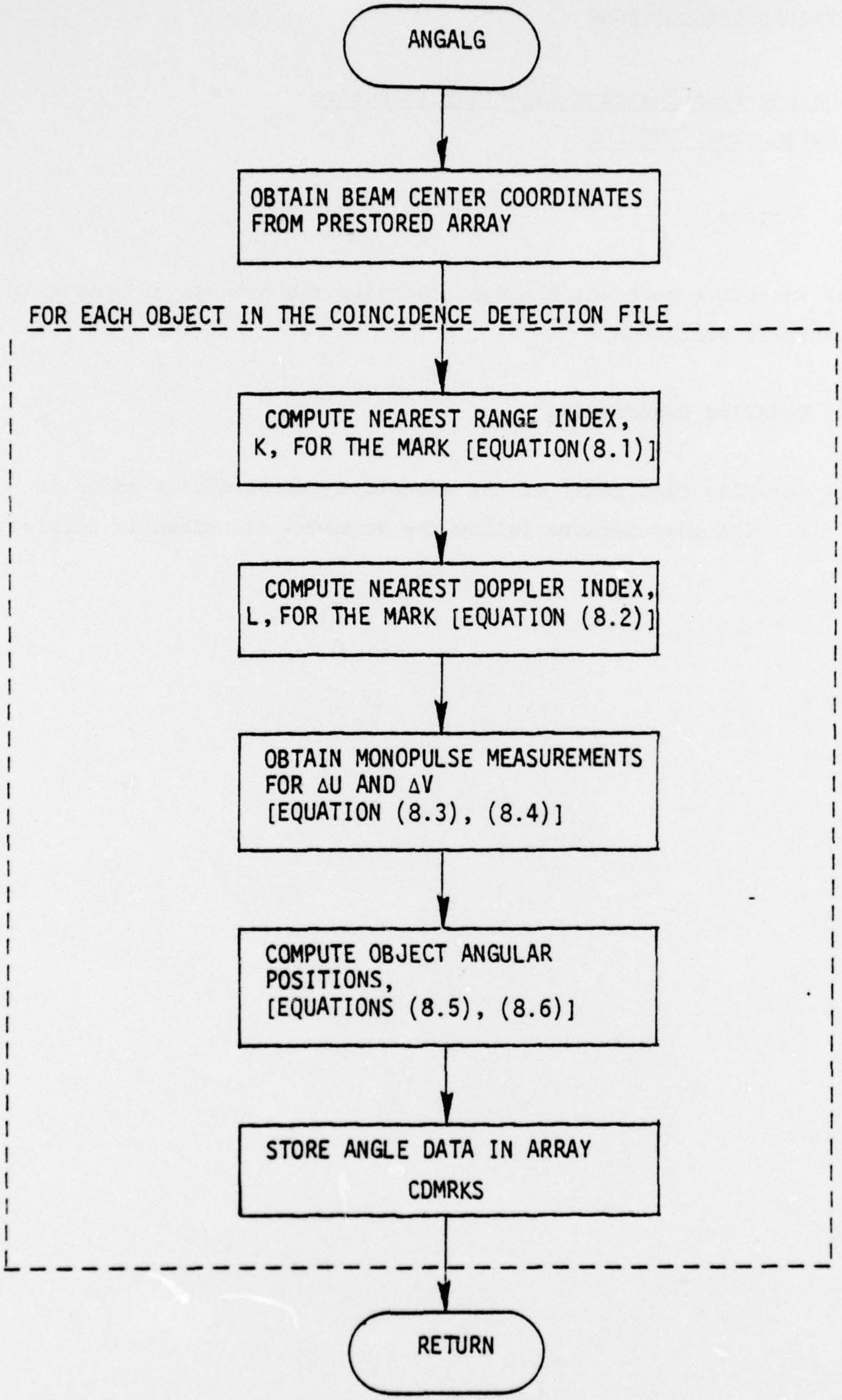


Figure 8.2 Object Beam Position Logic Detailed Flowchart

9.0 KNOWN OBJECT RECOGNITION ALGORITHM

9.1 PURPOSE OF SUBSYSTEM

The function of the Known Object Recognition (KOR) Algorithm is to identify a target designated by the CDG algorithm as one that is already in track and thus avoid initiation of a new track on a previously designated target. The track state may have resulted from bulk filtering in the same beam or in a different beam in an earlier scan. KOR Algorithm is also responsible for identifying two designations of the same object, resulting from the bulk filtering of two adjacent beams, performed within less than 50 milliseconds of each other.

9.2 GENERAL DESCRIPTION

The KOR algorithm operates on two state estimates, one provided by the CDG Algorithm and the other by the Track Function (or by designation of an object in another beam). The covariances of the corresponding estimate errors are also available to the KOR. The algorithm consists of a test which decides whether the two state estimates (often corresponding to two different time instants) correspond to the same object or to two different objects. Clearly, this test should also use the covariance of the errors in the designation and track state estimateion.

The performance of any KOR algorithm is specified in terms of (1) the probability of deciding that the two state estimates correspond to the same object when this is in fact true and (2) the probability of deciding that the state estimates correspond to the same object when, in fact, they correspond to two different objects. While the latter probability should be kept very small, the former has to have a reasonably high value so that there is a low probability of initiating a redundant track.

An overview flowchart for the KOR Algorithm is shown in Figure 9.1.

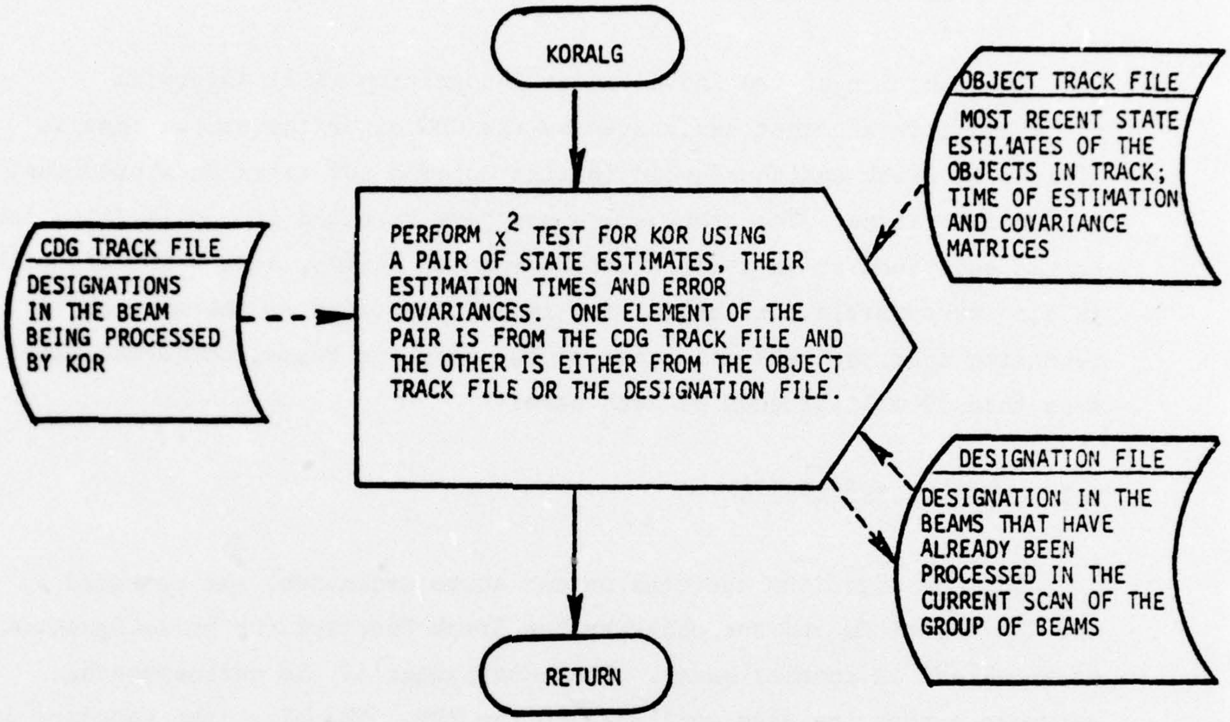


Figure 9.1 Overview Flowchart of the KOR Subsystem

An overview flowchart for the KOR Algorithm is shown in Figure 9.1.

When the designation sequence in a beam is completed, the KOR Algorithm is called. The input to KOR consists of the designations made up to that point, in the current scan of the beam group and the state estimates provided by the track function.

Each one of the new state estimates in the beam being processed is paired up with every one of the designations from the adjacent beams and the track state estimates.

In Figure 9.1 the OBJECT TRACK file contains the track state estimates along with estimation covariances and estimation times of all of the objects that are being tracked.

The DESIGNATION file contains the designations (states and covariances) accepted in the current scan. Each designation entry contains a reference to the beam index in which the estimate was generated. The KOR algorithm uses all of the state estimates provided by the track function and only those estimates from the designations of the current scan that correspond to the beams adjacent to the one being processed by KOR. This is why track estimates are stored in an object-oriented file while the current scan designations are stored in a beam-oriented file.

From the new designations in the beam being processed, only those that are accepted by the χ^2 test as new objects are retained and added to the list of the designations accepted in the current scan. The rest are dropped.

9.3 DATA SPECIFICATION

9.3.1 Input Data

The following data is input from the CDG Algorithm via the CDG Track file:

Program Symbol	Engineering Symbol		Description
NTN	NT_N		Number of objects designated in the current beam by the CDG Algorithm after N transmissions
CDGTRK(K,M)	$DT_N(n,m)$		Range-ordered designation list, indexed by k.
	$\chi_A(1)$	m = 1	Range estimate (m)
	$\chi_A(2)$	m = 2	Range-rate estimate (m/sec)
	M	m = 3	Number of misses
	$P_A(1,1)$	m = 4	Variance of range estimate (m^2)
	$P_A(1,2)$	m = 5	Covariance of range and range-rate estimates (m^2/sec)
	$P_A(2,2)$	m = 6	Variance of range-rate estimate (m^2/sec^2)
	$\chi_A(3)$	m = 7	Angular coordinate U
	$\chi_A(4)$	m = 8	Angular coordinate V

The following data is input via the DESIGNATION file:

Program Symbol	Engineering Symbol		Description
NDES	-		Number of objects in the array DESIG
DESIG(N,M)	-		List of objects, indexed by N, designated in the beams already processed in the current scan of the group of beams.
	$\chi_B(1)$	m = 1	Range estimate (m)
	$\chi_B(2)$	m = 2	Range-rate estimate (m/sec)
	$P_B(1,1)$	m = 3	Variance of range estimate (m^2)
	$P_B(1,2)$	m = 4	Covariance of range and range-rate estimates (m^2/sec)
	$P_B(2,2)$	m = 5	Variance of range-rate estimate (m^2/sec^2)

9.3.1 (Continued)

$X_B(3)$	m = 6	Angular coordinate U
$X_B(4)$	m = 7	Angular coordinate V
IU	m = 8	U index for beam of designation
IV	m = 9	V index for beam of designation

The following data is input from the TRACK FUNCTION via the OBJECT TRACK file:

NTRK		-	Number of objects in the array TRACKS
TRACKS(N,M)		-	List of the most recent data on the objects in track, indexed by K
	$X_B(1)$	m = 1	Range estimate (m)
	$X_B(2)$	m = 2	Range-rate estimate (m/sec)
	$P_B(1,1)$	m = 3	Variance of range estimate (m^2)
	$P_B(1,2)$	m = 4	Covariance of range and range-rate estimates (m^2/sec)
	$P_B(2,2)$	m = 5	Variance of the range-rate estimate (m^2/sec^2)
	$X_B(3)$	m = 6	Angular coordinate U
	$X_B(4)$	m = 7	Angular coordinate V
		m = 8	Time of estimation (msec)

The following data is input via the KOR PARAMETERS file:

GAMMA	γ	A positive real number specifying the threshold of the χ^2 test performed for KOR.

The following data is input via the BEAM STATUS file:

ICUR	-	Index to array entries for current beam
IU(ICUR)	Iu	Index for beam U coordinate
IV(ICUR)	Iv	Index for beam V coordinate
TCDG(ICUR)	T_{CDG}	Designation time for current beam (msec)

The following data is input via the MEASUREMENT PARAMETERS file:

SIGUV	σ_u, σ_v	Angle measurement standard deviation for U and V

9.3.2 Output Data

The output data consists of an updated DESIGNATIONS file for the beam group. Nonredundant designations from the current beam are added to the file by the KOR Algorithm. The data is passed to the Track Function and represents the principal output of the ALF.

9.3.3 Parameter Settings

The only parameter that should be specified is the threshold γ . Normally the value of γ is chosen such that when the two state estimates used in the KOR test correspond to two different objects and the spacing of the two objects is equal to or larger than the average resolution of the designation and track functions then the probability of deciding that the two objects are one object is very small. If this leakage probability is assigned a value of .005, the appropriate γ value is 4.7.

9.4 MATHEMATICAL RELATIONSHIPS

The test for KOR is a χ^2 test of hypothesis. Let χ_A and χ_B represent the pair of state estimate vectors for which the test is to be performed. Let P_A, t_A, and P_B, t_B be their error covariance matrices and times of estimation, respectively.

The test of the hypothesis that the two state estimates correspond to the same object consists of comparing the quadratic form

$$Z \triangleq (\chi_B - \Phi(t_B, t_A)\chi_A)^T Q_o^{-1} (\chi_B - \Phi(t_B, t_A)\chi_A) \tag{9.1}$$

with the threshold γ . In the expression (9.1), we have

$$Q_o = P_B + \Phi(t_B, t_A) P_A \Phi^T(t_B, t_A) \tag{9.2}$$

where $\Phi(t_B, t_A)$ is the state transition matrix which is uniquely determined by $t_B - t_A$ (only constant velocity trajectories are assumed). Therefore, we have

$$\Phi(t_B, t_A) = \left[\begin{array}{c|c} I & (t_B - t_A) I \\ \hline 0 & I \end{array}\right] \tag{9.3}$$

where I is the identity matrix. If

$$Z > \gamma \tag{9.4}$$

it is decided that the two state estimates correspond to two different objects and hence the estimate from the new designation is retained. Otherwise, the two estimates are assumed to correspond to the same object and the new designation is dropped. The value of γ is a fixed input parameter.

9.5 DETAILED DESCRIPTIONS

9.5.1 Executive Subroutine

9.5.1.1 Purpose

The EXECUTIVE SUBROUTINE is basically responsible for reading the appropriate data from the input files and extracting the pair of designations which are passed to the subroutine CHITST. It also prepares the information that is used to update the DESIGNATION file.

9.5.1.2 Detailed Flowchart

The flowchart of KORALG shown in Figure 9.2 is self explanatory. Note that since square beam packing is assumed (in u-v space) and the beams are scanned each row at a time in the direction of increasing v, the only beams adjacent to (IU,IV) for which the designation sequence is completed are (IU-1, IV-1), (IU-1,IV), (IU-1, IV+1) and (IU, IV-1). Each time the χ^2 test for KOR decides that the two designations (one of them is from the CDG TRACK file) correspond to the same object, that designation is deleted from the CDG TRACK file. A check is made to see whether the list has been diminished to an empty list, in which case control is returned to the ALF Executive.

When the CDG TRACK file has been tested against all designations in adjacent beams and all OBJECT TRACK file state estimates, the remaining entries in the CDG track file are transferred to the DESIGNATION file. This process updates the DESIGNATION file for use on the next execution of the KOR Algorithm.

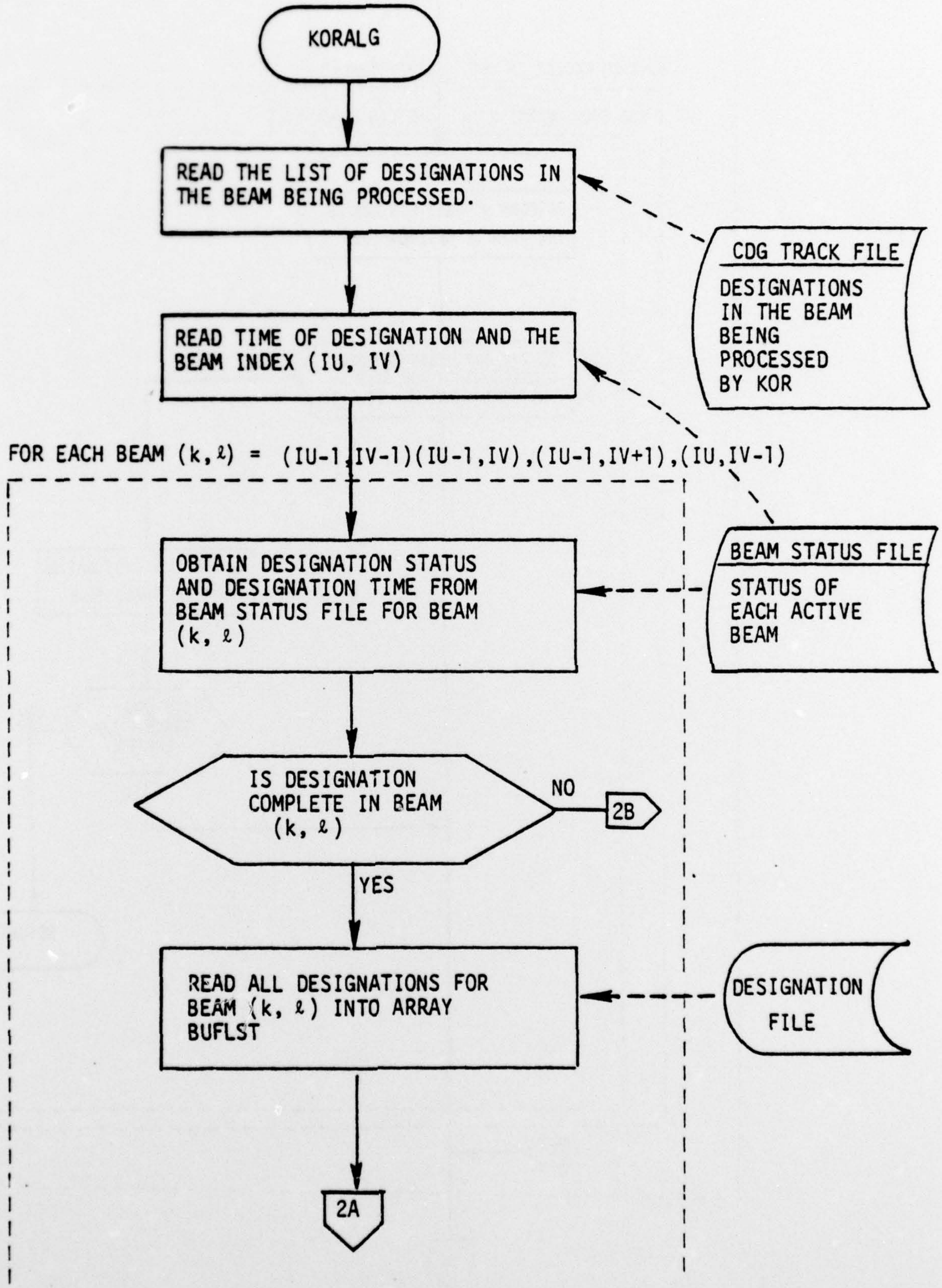


Figure 9.2 Known Object Recognition Algorithm Detailed Flowchart

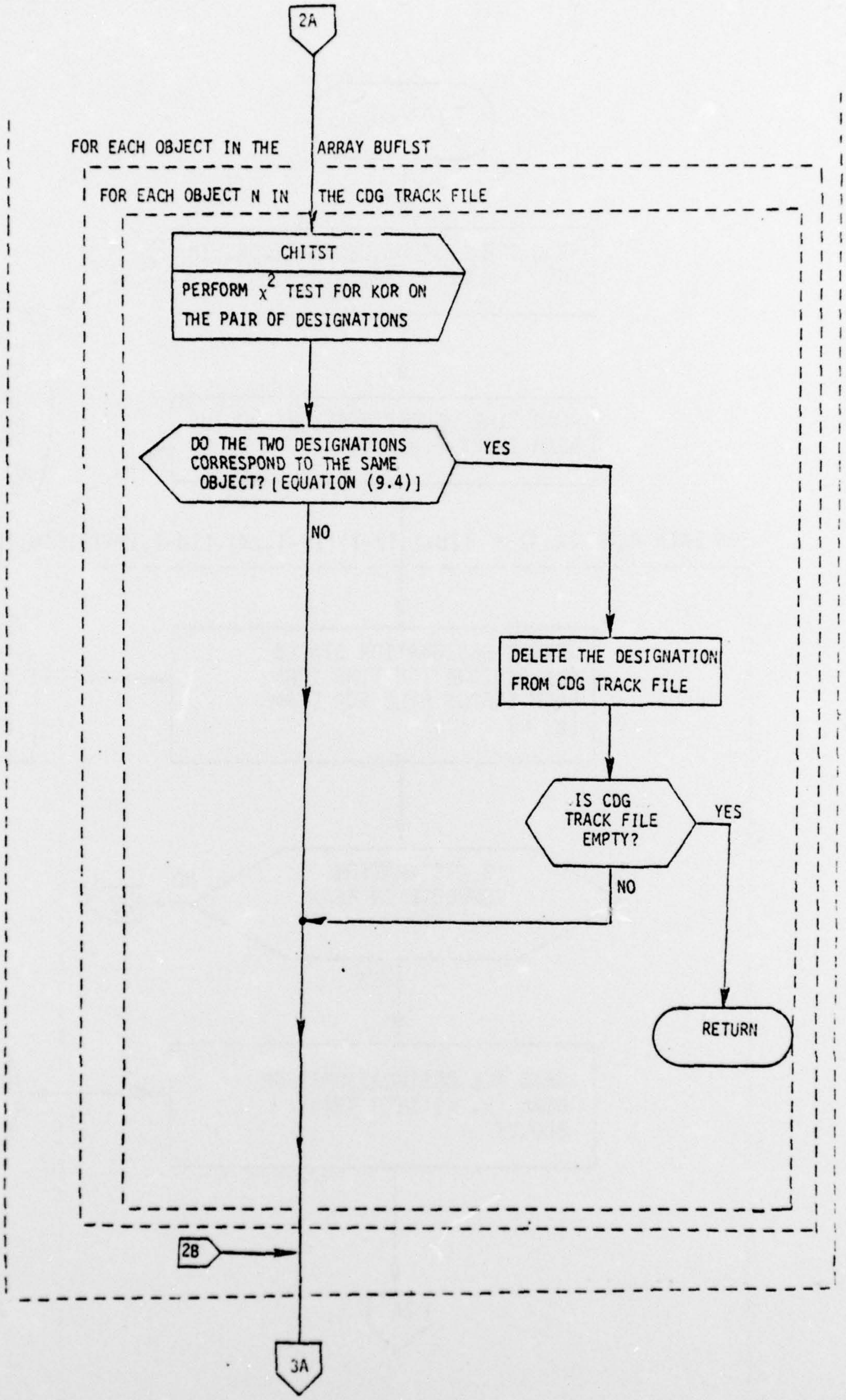


Figure 9.2 Known Object Recognition Algorithm Detailed Flowchart (Continued)

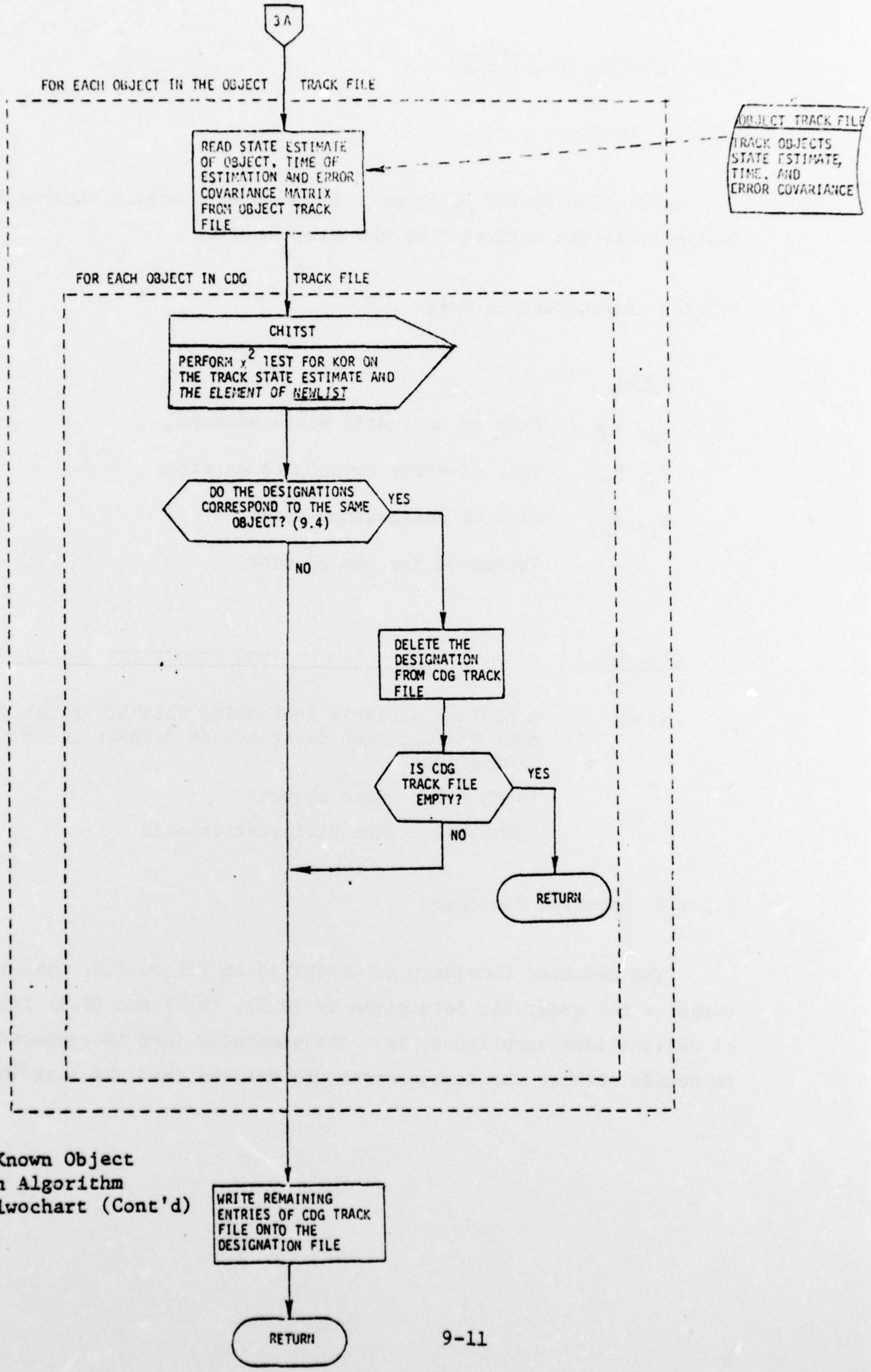


Figure 9.2 Known Object Recognition Algorithm Detailed Flwochart (Cont'd)

9.5.2 Subroutine CHITST

9.5.2.1 Purpose

Subroutine CHITST performs a χ^2 test to determine whether two designations are derived from the same object.

9.5.2.2 Input/Outuput Data

Input

X_A, X_B	Pair of estimated state vectors
P_A, P_B	Pair of error covariance matrices
t_A, t_B	Pair of designation times
γ	Threshold for the χ^2 test

Output

ISAME — A Boolean variable indicating whether or not the test decided that designations correspond to the same object

ISAME = 1	Same object
ISAME = 0	Two different objects

9.5.2.3 Detailed Flowchart

The detailed flowchart of CHITST is in Figure 9.3. The subroutine computes the quadratic form given by (9.1), (9.2) and (9.3) for the pair of designations supplied to it. The quadratic form is compared with γ to decide whether the designations are derived from the same object.

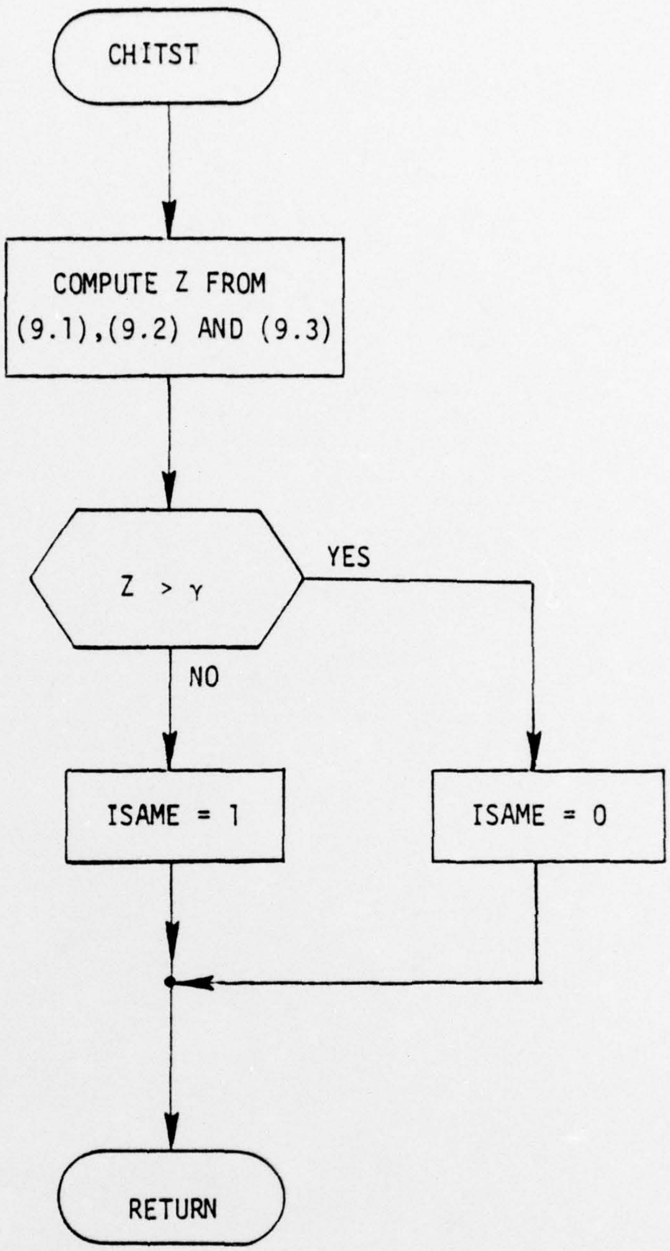


Figure 9.3 Subroutine CHITST Detailed Flowchart

10.0 MEMORY REQUIREMENTS

10.1 PURPOSE

This section provides estimates of memory requirements for the ALF executive, component algorithms, and large arrays of data. These estimates are intended only as initial guidelines for development programmers when allocating memory resources.

10.2 PROGRAM SIZE ESTIMATES

Table 10.1 lists estimated program size for the ALF executive program and each of the component algorithms. The term "program size" means words of instructions and internal data, such as parameters, pointers, small arrays, etc. The term "program size" does not include large arrays of data, such as radar returns, etc. Large arrays are described in Section 10.3.

Some of the program size estimates were obtained from existing FORTRAN programs implemented on the CDC 7600. These programs are marked by an asterisk in Table 10.1. The estimates based on actual programs may be excessively large, since each program was developed for use in a stand-alone mode, i.e., as a separate execution. Therefore, the development of compact code was not a primary consideration.

Those program size estimates not derived from existing programs were obtained based on relative algorithm complexity as compared to existing programs.

Table 10.1

PROGRAM SIZE ESTIMATES FOR ALF

	Program Size (Decimal Words)
Executive Program	1000
Waveform Request Algorithm	500
Parameter Selection Algorithm	250
Marking Algorithm*	2500
Coincidence Detection Algorithm*	250
Coherent Double Gating Algorithm*	1000
Object Beam Position Algorithm	500
Known Object Recognition Algorithm	500
TOTAL ALF SIZE	6500

*Based on actual implementations

10.3 ARRAY SIZES

Table 10.2 contains size estimates for the large arrays of data used by ALF. The file names and array names correspond to the terminology used in the Data Specification subsections for each algorithm. The values listed for Estimated Array Sizes are based on the assumed dimensions shown in the second column of Table 10.2. These dimensions are preliminary estimates and may require adjustments based on operating experience with the ALF simulation.

Some of the data files listed in Table 10.2 must be retained during the time between processing successive transmissions for an active beam, i.e., a beam undergoing designation. Data from other beams will normally be processed during this time. The fourth column of Table 10.2 lists the total number of versions of each file which must be retained between ALF executions. For example, the BURST 1 MARKS file for an active beam must be saved for use by the Coincidence Detection Algorithm when processing burst 2. After completion of burst 2 processing, there is no further need to save the BURST 1 MARKS or the BURST 2 MARKS data.

Some type of file management scheme may be required to prevent the accumulation of large numbers of sizable, unnecessary data files during a multi-beam execution of the ALF.

The item NBM in Table 10.2 represents the maximum number of beams which will undergo designation processing during a simulated attack. This parameter is highly dependent on scenario and beamwidth, and is not specified in this report.

Table 10.2

ARRAY SIZE ESTIMATES FOR ALF

File Name	Array Names and Dimensions	Estimated Array Sizes (Decimal Words)	Total Number of File Versions Saved
BURST 1 VIDEO	VIDEO1(2000,9) 2000 range bins 9 range-rate bins	18,000	One per active beam
BURST 2 VIDEO	VIDEO2(2000,9) Same as VIDEO1	18,000	None
BURST 1 MONOPULSE	UMONO1(2000, 9) Same as VIDEO1 VMONO1(2000,9) Same as VIDEO1	18,000 18,000	One per active beam
BURST 2 MONOPULSE	UMONO2 (2000,9) Same as VIDEO1 VMONO2(2000,9) Same as VIDEO2	18,000 18,000	None
BURST 1 MARKS	RRDOT1(800,3) 800 marks 3 data items per mark	2,400	One per active beam
BURST 2 MARKS	RRDOT2(800,3) Same as RRDOT1	2,400	None
COINCIDENCE DETECTION MARKS	CDMRKS(800,5) 800 marks 5 data items per mark	4,000	None
CDG TRACK	CDGTRK(600,8) 600 initial object 8 data items per object	4,800	One per active beam
BEAM DESIGNATIONS	DESIG(50xNBM,9) 50 designations per beam NBM active beams 9 data items per design.	450 x NBM	One per designated beam
WAVEFORM PARAMETERS	BEAMS(IU,IV,4) IU,IV defines 1500 beams 4 data items per beam	6,000	One
BEAM STATUS	IDES(NBM), TBURST(NBM), IU(NBM), IV(NBM), NBPAIR(NBM), NBURST(NBM) NBM active beams	6 x NBM	One
OBJECT TRACK	TRACKS (25 x NBM, 8) 25 objects per beam NBM active beams 8 data items per object	200 x NBM	One
OBJECT BEAM POSITION PARAMETERS	UBEAM(IU,IV) VBEAM(IU,IV) IU,IV defines 1500 beams	1500 1500	One